IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

GIRAFA.COM, INC.,

Plaintiff and Counterclaim Defendant

v.

AMAZON WEB SERVICES, LLC, AMAZON.COM, INC., IAC SEARCH & MEDIA, INC., SNAP TECHNOLOGIES INC., YAHOO! INC., SMARTDEVIL INC., EXALEAD INC., and EXALEAD S.A.,

Defendants and Counterclaimants.

C.A. No. 07-787-SLR

EXPERT REPORT OF SAUL GREENBERG, PH.D.

Table of Contents

I. INTRODUCTION AND SUMMARY OF OPINIONS

1. I, Saul Greenberg, submit this report on behalf of defendants Amazon Web Services, LLC, Amazon.Com, Inc., Alexa Internet, Inc., (collectively "Amazon"), IAC Search & Media, Inc. ("IACSAM"), Snap Technologies, Inc.("Snap"), Yahoo! Inc. ("Yahoo!"), Exalead, Inc., and Exalead S.A ("Exalead") (collectively "defendants") in this action. I understand that the plaintiff in this case has alleged that various defendants infringe some or all of claims 1,4-7, 12-18, 21-24, 29-35, 38-42, 44-46, 49-53, 55, 56 of United States Patent No. 6,864,904 (the "'904 patent"). I also understand that the defendants have challenged the validity of these same claims of the '904 patent. I have reviewed the '904 patent and its prosecution history.

2. It is my opinion that the alleged inventions described in the claims of the '904 patent are invalid because they are either anticipated by, or obvious in view of, the references discussed below, and detailed in the charts attached to this report.

3. It is my opinion that claims 44 and 55 are indefinite because one of ordinary skill in the art would not understand what is covered by these claims when read in light of the '904 patent specification.

4. It is my opinion that none of the asserted claims are entitled to the priority date of the provisional application.

5. Other opinions I have rendered are set forth in this report.

6. It is my understanding that discovery in this case is outstanding and ongoing. Specifically, it is my understanding that defendants contend certain documents have yet to be produced by Girafa, that certain responses to interrogatories are incomplete, and that

certain depositions have yet to occur. I reserve the right to supplement my report should this discovery, once received, impact my opinion in any manner.

II. SUMMARY OF QUALIFICATIONS

7. My educational background, professional achievements, and qualifications as a Computer Scientist and as an expert in Human Computer Interaction are detailed in my curriculum vitae, which is attached hereto as Exhibit A.

8. I earned a PhD in Computer Science from the University of Calgary in 1989; an MSc in Computer Science from U. Calgary in 1984; a Diploma of Education from McGill University in 1978; and a BSc in Microbiology and Immunology from McGill University in 1976. I am currently a Full Professor in the Department of Computer Science at the University of Calgary. I am also an Adjunct Professor in both the Department of Psychology at the University of Calgary and the Department of Computer Science at the University of Saskatchewan. I hold a joint National Science and Engineering Research Council ("NSERC") and an Informatics Circle of Research Excellence ("iCORE") Industrial Research Chair in the area of Interactive Technologies.

9. While my academic employment is based in Canada, the community I interact with as well as its literature is international. I am very familiar with work done in this area in the United States, frequently collaborate with colleagues located in the United States, and attend and give presentations at conferences based out of the United States, particularly conferences by the Association of Computing Machinery based out of New York.

10. I am an expert in Human Computer Interaction ("HCI"). Generally speaking, HCI is a discipline that covers the requirements, design, implementation and evaluation of computational systems for human use. I began graduate work in HCI in 1981. I continued research in HCI at the Alberta Research Council as an Industrial Postdoctoral Fellow starting in 1988, and then as a faculty member in academia starting in 1989. I was appointed as an Assistant Professor in the Department of Computer Science at the University of Calgary in 1990, and was promoted quickly through the ranks, receiving a Full Professorship in 1997. I have taught various introductory and advanced courses in HCI and related areas at the undergraduate and/or graduate level, including HCI, Computer Networking, Ubiquitous Computing, and Computer Supported Cooperative Work. My accumulated research contributions in these areas are recognized in the peer community through various awards, including: election to the ACM CHI Academy (2005), a University Professorship for Research Excellence (2005), and the Canadian Human Computer Communications Society (CHCCS) Achievement Award (2006). I have just been nominated by my University for a Royal Society of Canada Fellowship, which recognizes the merit and achievement of individuals; the decision will be made later this year. I have authored, edited and refereed a number of books, journals, conference papers, technical reports, videos and software manuals in the area of Human Computer Interaction, as detailed in Exhibit A.

11. As part of my research, I and several students directly under my supervision investigated how people access, revisit and represent web pages within a graphical user interface. This included the use of thumbnails and other methods of identifying web pages. I did this work both before and contemporary with the '904 patent, with some continuing to the present day. We note that [Cockburn-3], which I co-authored, is referred to on the face of the '904 patent,

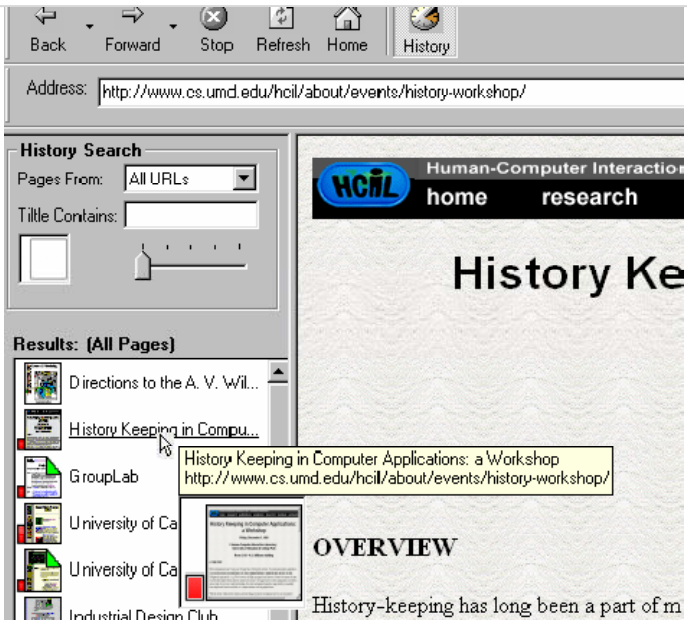


Figure II-1. Web History, from [Kaasten-1]

acknowledging our contribution and expertise in the domain of the patent. Brief summaries of my various projects in this area are described in the following paragraphs.

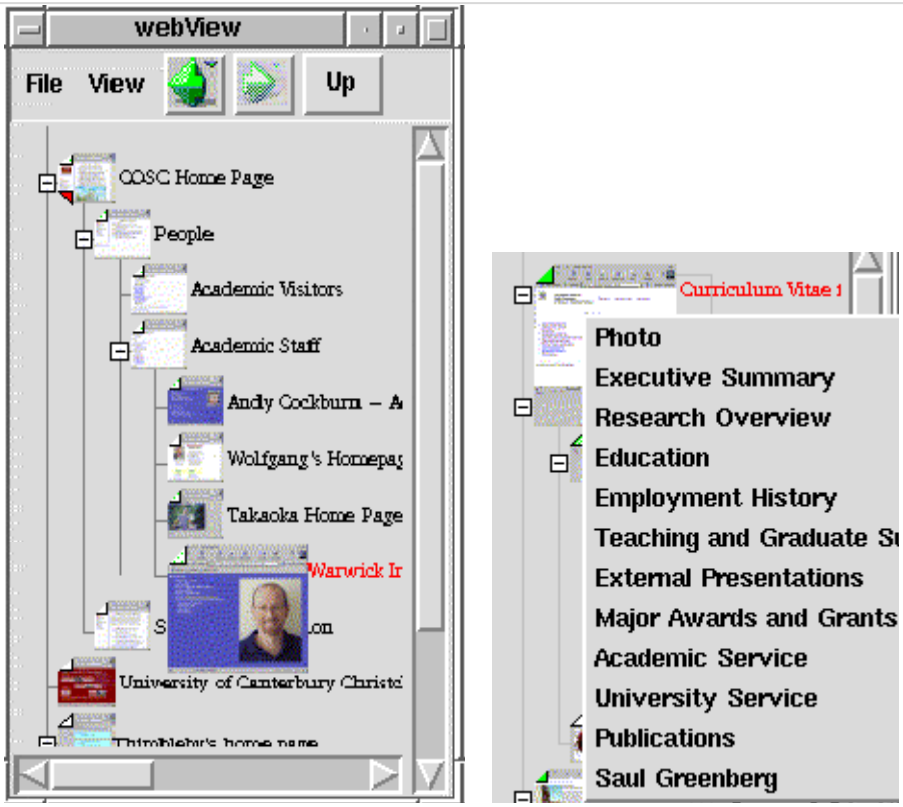
12. Around 1996, I and my graduate student Linda Tauscher examined how people access and revisit web pages by recording and quantitatively analyzing people's navigational patterns while using a web browser [e.g., Tauscher-1,2 & 3][1]. As part of this research, we offered a guideline recommending that interface designers should, when presenting a list of web items to the user, use a meaningful representation of a web page that includes semantic information about it. Various representations are suggested, including small thumbnails of the web page.

13. From 1999 – 2002, graduate student Kaasten and I published various papers exploring the use of thumbnail visual images in visual lists of web pages, where selecting a thumbnail would bring a person to a web page. Our papers, presentations and demonstrations presented a novel and publicly available system that we created and implemented. Figure II-1

[1] Citations enclosed in [] are found in Exhibit-B.

above, taken from [Kaasten-1], shows a sample screen shot of our system. The system captured information about pages a person had visited, which included rendering and storing a thumbnail visual image of each page. It then presented these annotated and searchable thumbnails in a graphical list that the end user could select from to rapidly return to those pages [e.g., Kaasten-1,2, 3 & 4]. One version of our system place this list in a sidebar integrated with the web browser, as visible in Figure II-1.This system also had a search filter that listed all previously-visited web domains, each represented by a thumbnail of the web page pointed to by that domain, (see in [Kaasten-2, Figure 2], reproduced later in Figure VI-13). [Kaasten-3] is a video illustrating the system features. Microsoft, Inc. directly supported this work through a grant, where we provided them with regular updates of it to them as it progressed. This system was made available to the public by our Grouplab Laboratory web site.

14. Roughly in parallel (circa 1999-2000), with New Zealand collaborator Professor Cockburn and others, we created WebView [Cockburn-1,2 & 3]. One of WebView's many features has it presenting web pages as a collapsible structural tree (a hierarchy) of visual thumbnails under its common root in the domain path (see Figure II-2 below, left side, taken from [Cockburn-1]). That is, all pages under a node could be collapsed and represented by that single entry and thumbnail [Cockburn-1]. Selecting that thumbnail with a left-button mouse press would raise the page in a browser. However, another feature displays all outgoing links contained by the page associated with the thumbnail within a selectable pop-up menu (see Figure II-2, right side, taken from [Cockburn-1]). When a user selected that thumbnail with a right-mouse button press, the popup menu with all alternate links are displayed; the user can then select any link to directly navigate to that page without having to visit the thumbnail's page. In



a) hub and spoke b) shortcut pop-up menu

Figure II-2. Webview, from [Cockburn-1]

this case, the thumbnail allows an end-user to select and browse to a page that differs from the page represented by the thumbnail. [Cockburn-2] is a video illustrating its features.

15. With graduate students Kaasten and Edwards, we examined and compared the recognizability of web page titles, URLs, and titles via a controlled user study, where we considered thumbnails of various sizes as a factor [Kaasten-5].

16. With graduate students Rounding and later McEwan, we let people post web pages of interest to a group: our groupware systems (called Notification Collage and Community Bar) included a ‘web page / web item’ that let a person select a web page and post it to the group. The item created and centrally stored a thumbnail image of that web page - including additional information such as the http link address, who posted it, etc. - and then transmit and display those

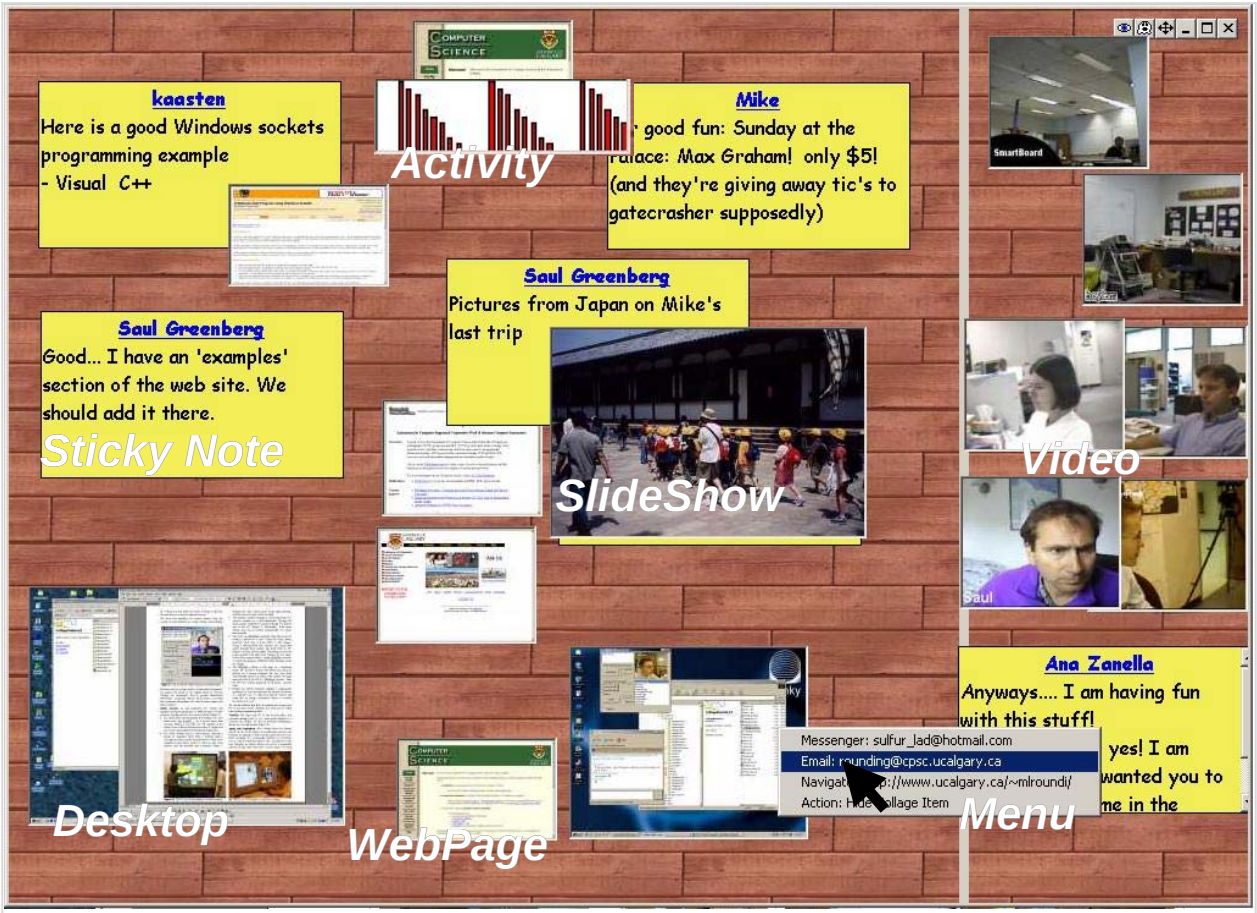


Figure II-3. Notification Collage showing the web item, from [Greenberg-1]

thumbnails on other people's computers. From that thumbnail, others could then review and decide to display that page. Figure II-3 above reproduces a screen snapshot from [Greenberg-1], with the 'WebPage' item visible as a thumbnail visual image at the bottom of the screen. Example publications illustrating this work include [Greenberg-1, McEwan-1].

17. With graduate student Boyle, we created a system where end-users interested in tracking a particular web page can specify that a notification of visual changes of that web page should be created; if the image contents of a page changes, the system then displays those changes in a viewer on the end-user's screen [Greenberg-2]. Our system allows the end-user to clip some or all of a web page image, and reduce it in size. By default, the entire page is used to produce a web page thumbnail. The system then polls that web page periodically

to see if its image has changed. If it has, it creates a new thumbnail image of the page and uploads it to a server (which behaves as an image server). A client viewer is notified that a new image is available, receives the new thumbnail from this server, and displays it on the screen. If the end-user clicks the thumbnail visual image, a browser appears containing the page it represents.

18. We have used visual thumbnails in various other non-web projects up until the present day. This includes: representing live working areas within a groupware system as miniatures (also known as radar views) so that others can see one's activities [e.g., Gutwin-1], and representing pages visited in a document as thumbnails attached to a scrollbar [Alexander-1].

19. As an undergraduate course instructor, I have taught undergraduate students how to build image web crawlers operating a multiplicity of downloaders (i.e., downloading in parallel) on lists of URLs, and how to construct simple image servers that return either images or pages containing a gallery of images to requesting clients.

20. As a professor and researcher, I have been involved in many service duties in the Human Computer Interaction domain that require me to evaluate the work (including its novelty) of other researchers. I have served as a referee for innumerable workshop, conference and journal article submissions. I am currently an editorial board member of three journals: Computer Supported Cooperative Work: The Journal of Collaborative Computing (Springer), the International Journal of Human Computer Studies (Elsevier), and Cognitive Technologies Journal (PMI). I have served as a member of many program committees for various academic conferences. I have served senior roles in these conferences. For example, in 2008-2009, I am currently the Papers/Notes co-Chair for the ACM Conference on Human Computer Interaction -

the major and largest conference in the HCI area - where I am in charge of managing the entire review process for over 1100 submitted papers. In 1998 and again in 2006, I was the Papers Committee co-Chair for the ACM Conference on Computer Supported Cooperative Work - a specialized conference in a sub-domain of Human Computer Interaction - where I was again in charge of the entire review process for all submitted papers. I regularly review academic grants submitted by other researchers to government granting agencies. I regularly evaluate other academics and professionals in the area in terms of their contributions to the field, e.g., for promotion, for academic and industrial positions, and for ability to carry out particular research projects. In 2001-2003, I was first a member and then the Chair of Canada's National Science and Engineering Research Council (NSERC) - Computing & Information Sciences Committee GSC-330. This committee evaluated and recommended granting levels for several hundred Computer Science grant submissions. In 2006-2007, I was a Panel Member of the United Kingdom's Engineering and Physical Sciences Research Council International Review of ICT Research within the UK, a panel that evaluated and made recommendations concerning the strengths, weaknesses, and future directions of Computer Science and Engineering research across the entire United Kingdom.

III. OTHER EXPERT TESTIMONY AND COMPENSATION

21. I have worked as an expert in various patent infringement cases and a trade-secret case involving conferencing systems, where my primary role was to search, investigate, analyze and opine on the prior art. Of these cases, I served as the testifying expert twice. First, I provided a deposition for Pixion, Inc. (Plaintiff) vs. Placeware, Inc. (Defendant) on behalf of the

defendant. Second, I provided a deposition for Cross Atlantic Capital Partners, Inc. (Plaintiff) vs. Facebook, Inc. and TheFacebook, LLC. (Defendants) on behalf of the defendants.

22. For the time spent on this matter, I am being compensated at my consulting rate of $450 per hour. Travel and other expenses are recompensed at cost. I have received no additional compensation for my work in connection with this case, My opinion is not influenced in any way by the compensation that I receive, and my compensation will not be affected in any way by the outcome of this case. To my knowledge, I have no other relationship or financial interest with either the Plaintiff or Defendants.

IV. INFORMATION CONSIDERED IN FORMING OPINIONS

23. My educational background, professional achievements, and qualifications in the fields of Computer Science and Human Computer Interaction have given me both a theoretical and an empirical grounding for assessing the validity of the claims of the ‘904 patent. The materials on which I have relied in drafting this report are listed in Exhibit B1. While not relied upon specifically in the drafting of this report, I considered additional materials while forming my opinion in this matter. This wide range of bibliographic materials which I considered is listed in Exhibit B2, I have also reviewed the ‘904 patent, its prosecution history and the US Provisional Patent Application No 60/169,328.

V. OVERVIEW OF THE '904 PATENT

24. The '904 patent states that its field of invention: "relates to Internet methodologies and systems generally and more particularly to systems and methodologies for displaying information received over the Internet" (1:12). To very roughly summarize, the '904 patent describes several embodiments and claims that specify how thumbnail visual images (including thumbnails of home pages) can represent and visually present hyperlinked world wide web or 'web' pages within a variety of layout schemes, where these images may be provided by an image server. Figure V.1 (displayed below) reproduces Figure 2 from the '904 patent, where it shows illustrates one embodiment as a simplified, partially pictorial partially block diagram. An example web page including these thumbnails is at its upper right.

25. In Paragraph 8, page 6 of the Memorandum Order Denying Preliminary Injunction (Document 242), the claimed inventive elements of the '904 patent are suggested to be limited as follows.

> "8. The prior art. There is no dispute that thumbnail visual images were well known in the art by December 1999, as were image servers. (D.I. 83 at ¶¶ 29-32; D.I. 240 at 27) According to the prosecution history of the '904 patent, the two allegedly novel or inventive elements of the patent are: (a) using thumbnail visual images of a homepage of a linked-to web site, instead of a thumbnail visual image of the linked-to web page; and (b) providing the thumbnail visual images via an image server that is separate from a web server . (D.I. 83, ex. B)

26. For the sake of summarizing the patent, the primary claimed inventive elements appear to be as follows. However, this list depends on how the court construes the various elements and claims of the patent:

- Displaying thumbnails of home page preview images, possibly including a requirement that the preview image not be of the linked to page;

- An image server for thumbnail images of web pages, operating separately from a web server;
- Splitting, trimming, and constructing a URL to produce 'the most representative image of a given web page'; and
- Using a multiplicity of downloaders to populate a database of thumbnail images of web pages.

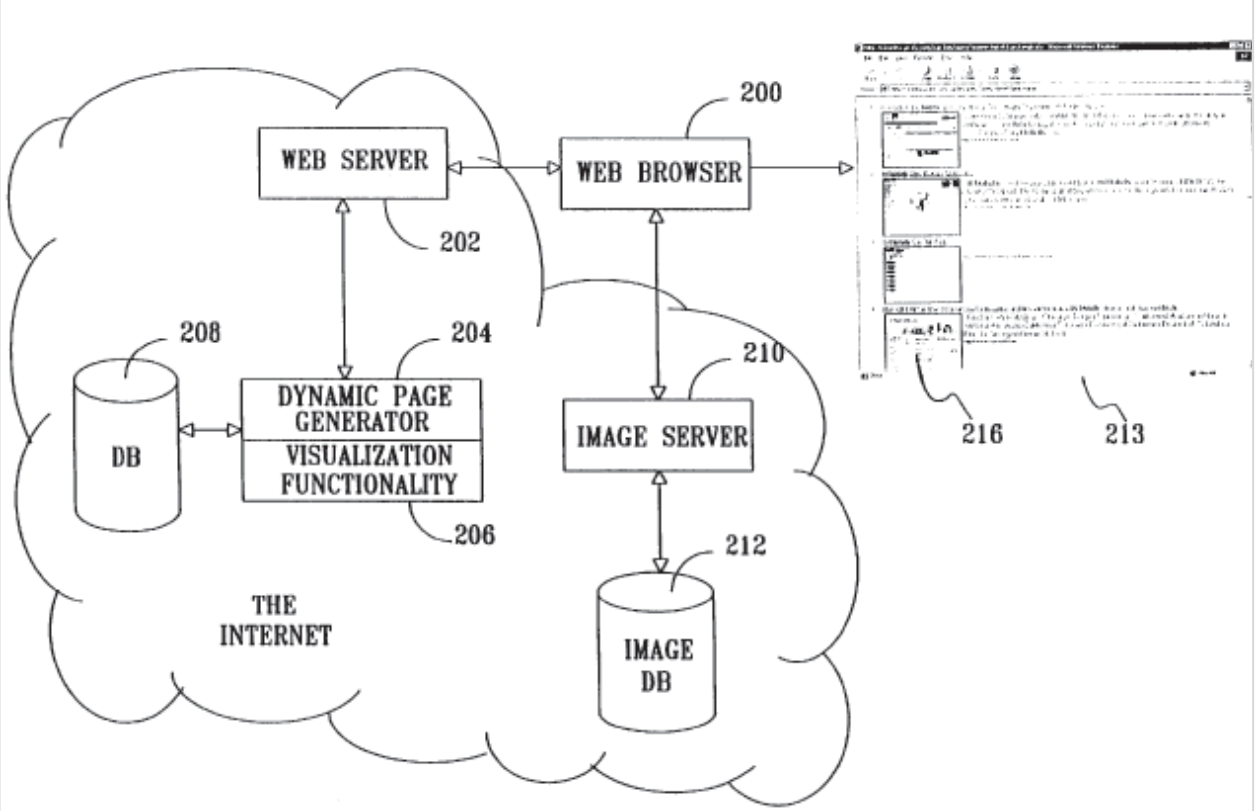


Figure V.1. Figure 2 from the '904 patent [US Patent 6,864,904]

VI. HISTORICAL BACKGROUND

27. As will be seen below, there is a rich history of research, development, systems and patents in this field, both before and contemporaneous with the '904 application. Indeed, the intellectual foundations behind many of the inventive elements claimed in the patent appeared well before the Internet and the World Wide Web, for example, in graphical user interfaces for desktop computers, in hypertext systems, and in distributed systems. This intellectual foundation cannot be ignored, for the Internet and World Wide Web of the late 1990s were greatly influenced by, emerged from, and often incorporated this prior knowledge.

28. I briefly review this history to set the context of knowledge prior to the '904 patent priority date. Among other things, this history verifies that not only were thumbnail visual images, home page thumbnails, and image servers separate from a web server well known in the art and used routinely, but that many researchers, developers and inventors were exploring highly nuanced uses of these ideas. Similarly, ideas such as trimming of URLs to construct new ones were in common use, as were web crawlers that operated a multiplicity of downloaders. Consequently, I roughly structure this history into four themes: A: thumbnail visual images as used to represent documents, pages, and home pages; B: image servers; C: splitting, trimming and constructing URLs, and D: multiplicity of downloaders.

A. THUMBNAIL VISUAL IMAGES

a. Vannevar Bush and Memex

29. The intellectual roots of the World Wide Web (WWW) are generally agreed by various academic communities to emerge from Vannevar Bush's 1945 article *As We May Think* [Bush-1]. Bush, then the Director of the Office of Scientific Research and Development

for the US Government, envisioned a system called *memex*. He described "a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility." Accessing information included known methods such as indexing. Yet Bush believed the essential feature of memex was associative indexing – what we would now called hypertext linking on the WWW – "whereby any item may be caused at will to select immediately and automatically another… the process of tying two items together is the important thing." Computers for everyday use did not exist in 1945. Thus Bush imagined his technology built on microfilm records. He described how microfilm – produced by microphotography – stores images 20x smaller than the original, and anticipated ratios of 100x in the future. He envisioned "a library of a million volumes could be compressed into one end of a desk" and that the entire store of human knowledge could fit into a small moving van. While the images are small, he saw that they could be projected onto a translucent screen for convenient reading. He also mentioned that such a device could presumably be operated from a distance, again foreshadowing location-independent access to information, as found on the modern Internet in general and World Wide Web specifically.

30. Bush's work foreshadowed and anticipated the use of thumbnail visual images described in the '904 patent 55 years before the patent filing date. In *memex*, the images are literal thumbnail visual images, for they are indisputably images 20 to 100 times smaller than the original sources. Furthermore, Bush placed no limit on how such microfilm thumbnails should be projected, i.e., they do not have to be seen in their original size. If seen full sized, an image is characterized as the document itself. If seen substantially smaller, the image matches how one would perceive thumbnails. Indeed, later devices for standard microfiche viewing included special apparatus for allowing a person to adjust the magnification of the image seen

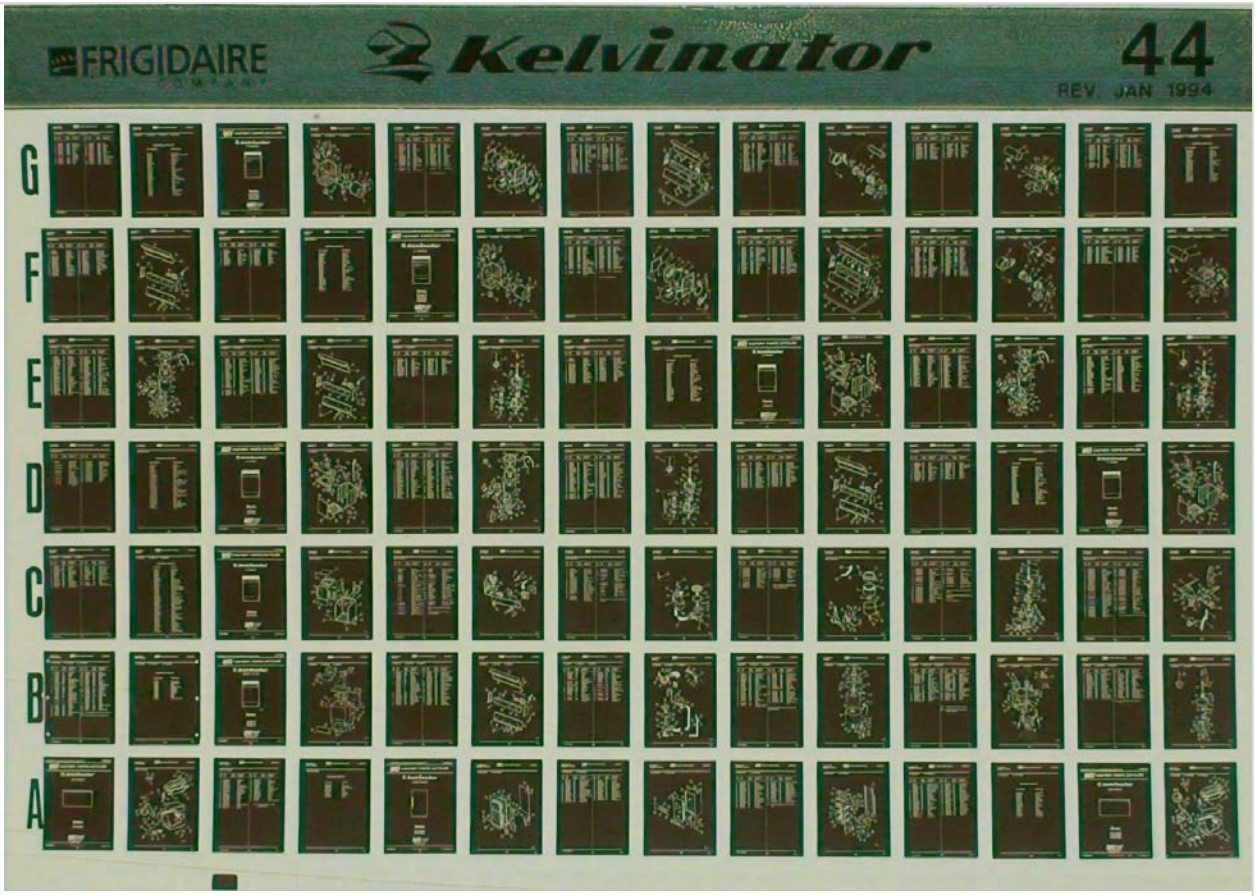


Figure VI-1. An example Microfiche from 1994, from http://upload.wikimedia.org/wikipedia/commons/e/e4/Microfiche_card.JPG

(e.g., see [US Patent 3,369,450, col. 6 line 35]. Thus one could presumably see stored microfiche images ranging from thumbnail size (as an overview of images on the microfiche) to full size. For example, consider the example microfiche dated 1994 in Figure VI-1 above. If the end-user zoomed the display out so that the entire microfiche occupied only a small part of the display in the microfiche viewer, then the image seen could be characterized as a thumbnail of a literal 'home page' that contained multiple selectable links to its children pages. As the person magnified the image, they would then see thumbnails of these children, where magnifying into a particular thumbnail would then reveal the page at full size.

b. Thumbnails of documents and images within a desktop browser or application: Feiner et. al.; Patent 5,060,135 (Levine); Wang Freestyle; Rao, et. al.'s Infogrid; Rao, et. al.'s Protofoil; Kullberg's Dynamic Timelines, Lowe et. al.'s Image Engine

31. Over time, visual thumbnails became a very well-known technique in desktop graphical user interfaces for providing previews of many types of digital information. Thumbnails were often implemented within desktop browsers and / or specialized applications. For brevity, I will only touch upon only a very few examples here, where I illustrate how visual thumbnails were known as a general technique in graphical user interface prior to the '904 filing date and used for visualizing digital data such as documents, pictures, and medical images. In later sections, I will focus on thumbnail use on the web.

32. In 1982, Stephen Feiner et. al. from Brown University described an experimental system for creating and presenting interactive documents [Feiner-1]. The system was a hypertext system, where documents could be connected by links. Thus it has characteristics resembling the links and hypertext properties of the World Wide Web. Many part of their system use thumbnail visual images of pages, which they call 'miniatures', to give end users a preview of document contents. When one clicks on a thumbnail, the full sized page is displayed. For example, the caption of Figure 8 in [Feiner-1], reproduced below in Figure VI-2 below reads

> "Fig. 8. The index page. All pages associated with a user-selected keyword are displayed. Colored bands, representing the pages' parent chapters, are alphabetically ordered by chapter name. They contain alphabetized page minatures, each of which accesses its page when touched. [Feiner-1, page 72]".

Similarly, Feiner et. al. incorporates thumbnails into their 'timeline page' that show recently viewed pages as thumbnails in chronological order, and a 'neighbours page' that shows thumbnails of a page and the surrounding 'neighbours' that link to and from it.

Figure VI-2. The Index page showing thumbnails, from [Feiner-1, Fig. 8]

33. In 1988, Levine et. al. from Wang Laboratories filed a Patent 5,060,135. Amongst other things, their system provided a graphical user interface to a data processing system based on the metaphor of a simulated desk, where documents were represented on the screen as thumbnail images (which they called ‘stamps’). They write:

> In a preferred embodiment, the desk view (i.e. the screen view of the user's system desk) shows a reduced image or "stamp" of one page of each document on the user's system desk or computer work area.
> Preferably the representations of documents are miniaturized or reduced images of one page of the document. The reduced images are referred to as stamps and each stamp serves as a unique direct representation, that is, an actual image or pictorial likeness of a certain document on the user system desk … the term "document" means various sheets of information whether a single page or multiple page document

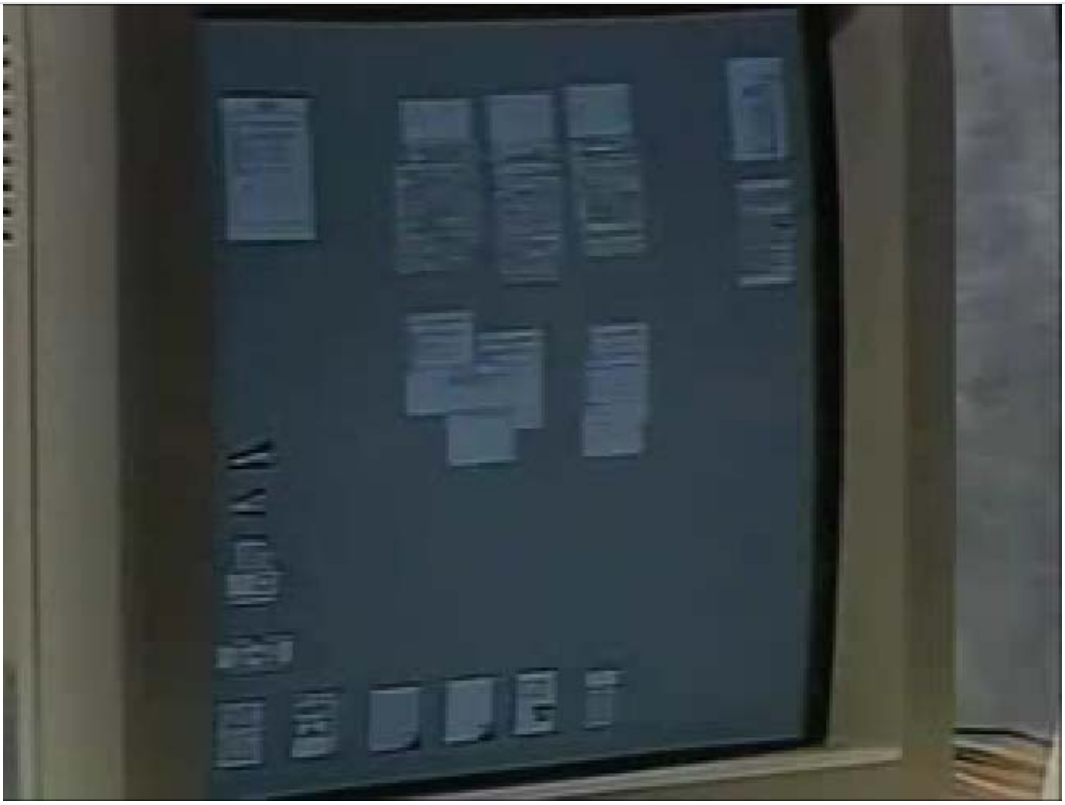


Figure VI-3. Freestyle, from [Wang-1, at time 6:54]

is involved… [US Patent 5,060,135, col. 3:28]

Levine et. al. go onto to describe how their thumbnail visual images, or stamps, are formed by various image reduction scheme for both black and white and color images [US Patent 5,060,135, col. 3:49-29]. Wang Laboratories also built and commercialized a system called 'Wang Freestyle'; they produced a video which illustrates how these thumbnails were generated and used within it [Wang-1]. At time 6:50, for example, the narrator illustrates how thumbnails are used on the Wang desktop (Figure VI-3 above reproduces a frame from the video showing the desktop at that time), where he says

> "The desk is a model of the way I do things here. In my in-basket you'll notice a miniaturized spreadsheet. In fact, what these are [pointing to thumbnails] examples …. Miniaturized versions of the page. Notice that you can kind of read them. And that's a special Wang advantage. Nobody else does that. We have a special algorithms [sic] and special routines for doing that." [Wang-1, time ~6:50]

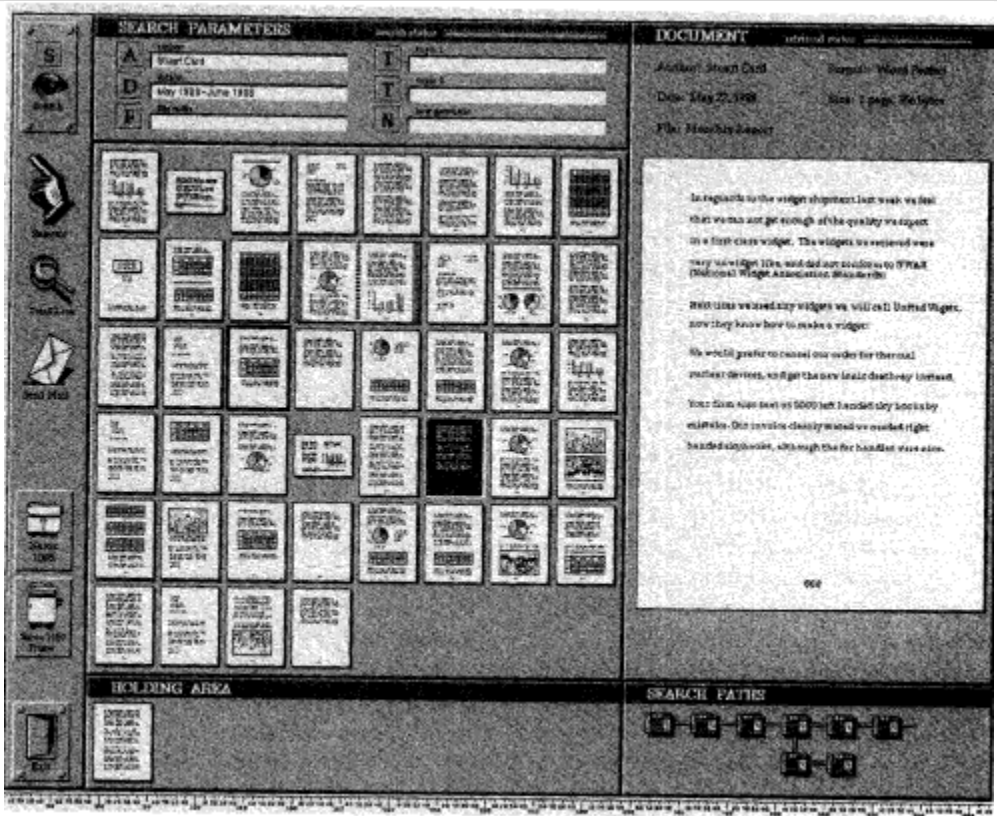


Figure VI-4. InfoGrid, from [Rao-1, Fig. 3]

Later, he shows how these thumbnails can be manipulated on the desktop, selected to open them at full size, and stapled together with other thumbnails.

34. In 1992, Rao, Card et. al. from Xerox PARC, published and presented a paper titled 'The Information Grid: A Framework for Information Retrieval and Retrieval-Centered Applications' [Rao-1]. They presented a framework called the InfoGrid, whose purpose was "to provide a user interface design and a retrieval-centered interaction model for information access applications", where "the design was initially targeted at providing an easy-to-understand-interface for managing documents stored as digital images in a distributed, multiuser, document database" [Rao-1, page 23]. Central to their user interface was the idea of selectable document thumbnails. As seen in Figure VI-4 above, which reproduces Figure 3 in [Rao-1], the large central part of the screen (which they name the 'Thumbnails Area' is a "browser area where the

system displays a visualization of the documents that matched the query", and the right side is where "Individual documents can be viewed in Document Area, which contains an image or textual view of the document's contents…" [Rao-1, page 24]. In the quote below, they describe how each thumbnail returned by a query is generated from the first page of the document. Thus this thumbnail of the first page is similar to a 'home page' that is the entry point to the other pages contained by that document:

> "The results of the query are displayed in the results area as some kind of visualization. Currently, retrieved documents are shown as fixed-size thumbnail sketches that are generally laid out in a grid (hence} the name InfoGrid). Thumbnails are generated for a document by scaling the first page down to the required size." [Rao-1, page 26]

Rao et. al. describe how the InfoGrid concept can be applied to a broad variety of situations. Examples include: an 'Electronic File Cabinet' to help a person access a store of scanned documents; a 'Network Document Database' which is similar except now the document store is distributed across a network; a 'Biography database' which allows retrieval of pictures and biographies of people; and an 'Encyclopedia' which allows retrieval of text entries from an encyclopedia.

35. In 1994, Rao, Card et. al. from Xerox PARC, published and presented another paper describing their Protofoil system [Rao-2], another system for "supporting document imaging for ad hoc information work and in demonstrating how a variety of user interface and information retrieval techniques can be integrated and applied" [Rao-2, page 180]. Figure VI-5 below, which reproduces Color Plate 2 from Rao-1, illustrates again how thumbnail visual images of documents (shown on the right) are central to this user interface. In this case, they define the right thumbnail view as a user interface to browsing documents returned by

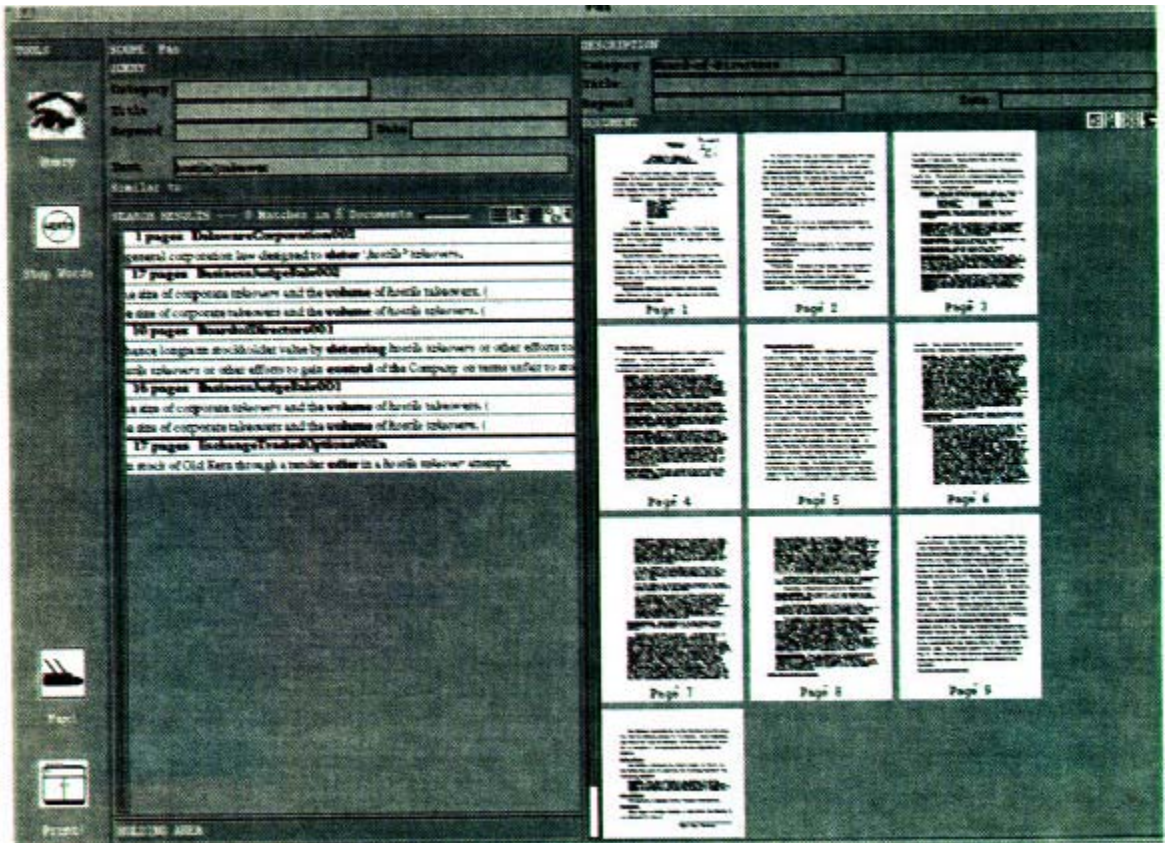


Figure VI-5. Protofoil, from [Rao-2, Fig2]

retrieval requests and search results [Rao-2, page 182]. They also describe how these thumbnail search results can be generated from five search methods [Rao-2, page 183].

36. In 1995, Lowe et. al. from the University of Michigan published a paper [Lowe-1] that described the Image Engine, "a prototype microcomputer-based system for the storage, retrieval, integration, and sharing of a wide range of clinically important digital images" [Lowe-1, page 57]. The images they describe are those produced by imaging technology in clinical medicine, such as traditional radiological and nuclear medicine images, CT tomography,MRI magnietic resonance imaging, PET positron emission tomography, ultrasonography, and others [Lowe-1, page 57]. Their system included a separate image browser that displayed various images as thumbnail visual images. Figure VI-6 below reproduces Figure 2 from [Lowe-1], which is a screen snapshot of the Image Engine Thumbnail Browser. Their text

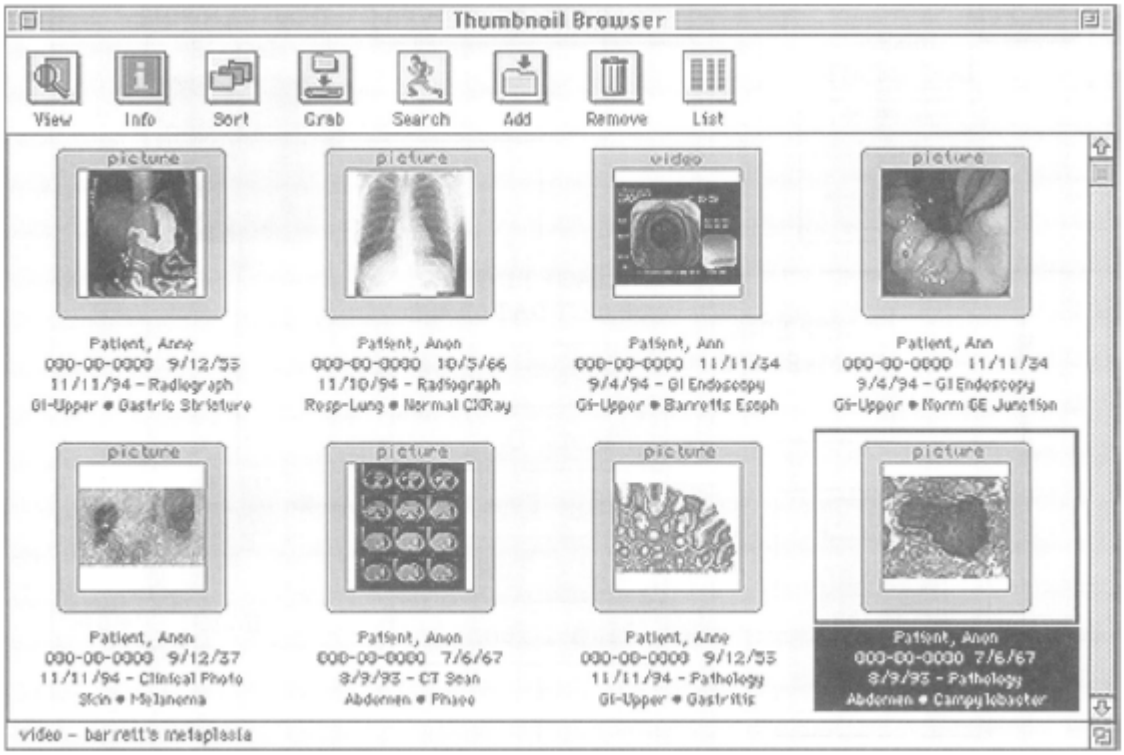


Figure VI-6. Image Engine Thumbnail Browser, from [Lowe-1, Figure 2]

anticipates search results that returns images as thumbnails from a separate image server, where selecting a thumbnail would then show the image at a larger size:

> The user may search for image subsets using combinations of image properties, including Metathesaurus terms. Retrieved subset summaries of images can be viewed as either a scrollable list of thumbnail images (100 by 100 pixel scaled, 24-bit color images) or a text list of object identifiers (image name, type, patient ID, and so forth). Images can be selected and viewed at full or scaled size on the computer display in resizable, scrollable windows. Multiple images can be viewed simultaneously. Image information text can be viewed simultaneously with images. Retrieved image sets can be sorted and displayed on a number of criteria. [Lowe-1, page 60]

37. In 1996, Robin Kullberg, from the MIT Media Laboratory demonstrated another use of thumbnails to display photographic images in the "Dynamic Timelines" system that displayed a history of photography [Kullberg-1]. She says in the video: "this history of photography presents a database of over 200 annotated photographs from the collection of the George Eastman House. Upon first view, the user is presented with an overview in which the

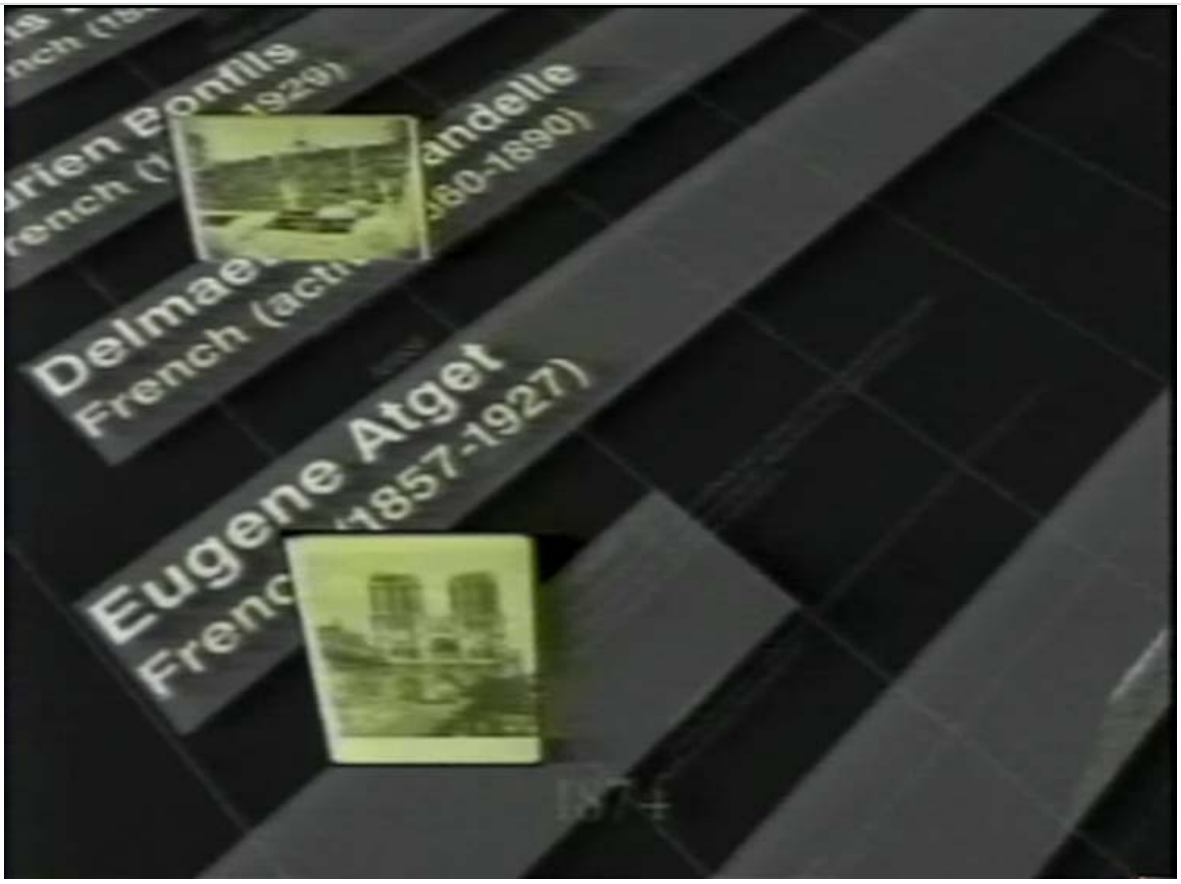


Figure VI-7. Dynamic Timelines, from [Kullberg-1, Time 1:18]

database is spread out before here as a 3 dimensional landscape of information. …". The end user can navigate around the 3 dimensional space, zoom in to see these images at various sizes, and select one to see it at its full size. A single frame from her video, taken at time 1:18, is reproduced in Figure VI-7 above.

c. Thumbnails of web pages within desktop browsers: Visioneer's Visual Explorer, and Internet Explorer 4

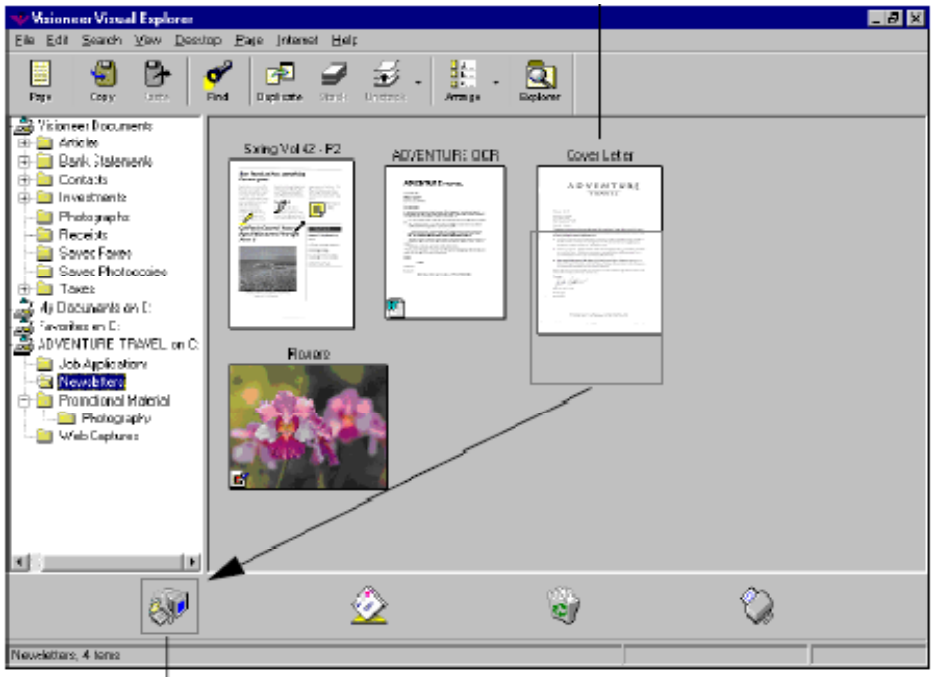
38. As the World Wide Web gained popularity in the 1990s, various systems incorporated thumbnail visual images of web pages into the desktop browsers and/or specialized applications. These thumbnails were typically in addition to the other document types such systems already displayed, as described in the previous section. That is, displaying visual

thumbnails of web pages was a natural evolution of displaying visual thumbnails of other document types. To illustrate, a few examples are described here.

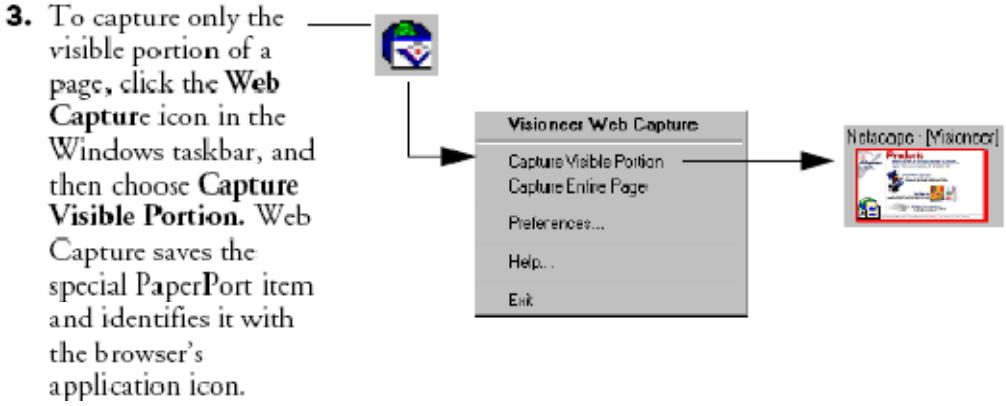
39. In 1998, Visioneer described their Visioneer Visual Explorer 1.0 in its 'Visioneer Visual Explorer Getting Started Guide for Windows'. They illustrate a system that fuses the concept of showing thumbnail visual images of traditional computer documents to include thumbnail visual image of web pages. As part of their system, they show several ways – including thumbnail visual images – to view a variety of items (e.g., documents, images, web pages) on the graphical user interface to the Windows operating system. They write:

> "Desktop View displays a thumbnail, a small graphic that represents each item in a folder. The thumbnails represent PaperPort (max files) being the format in which Visual Explorer saves items. The thumbnails can also be non-PaperPort items – times created using other applications, such as Microsoft Word. These items can be located anywhere on your computer". [Visioneer-1, page 2]

Each thumbnail links to the actual document, where double-clicking the thumbnail raises the full sized document into a separate application called the Page View [Visioneer-1, pages 2-3]. These thumbnails are visible in Figure VI-8 below at its top, which reproduces the screen snapshot from [Visioneer-1, page 9]. The Visual Explorer also allows an end user to create stacks of items, a single front thumbnail represents all items in the stack, although users also have the option to navigate through thumbnails of other pages under this stack [Visioneer-1, page 6]. One skilled in the art would know that this is similar to a home page that represents a sequence of pages under it, and that the ability to see multiple thumbnails shows a preview of those pages. Other parts of the User Guide illustrate how the system displays thumbnails of many other types of documents, including fax documents [Visioneer-1, page 9] and images [Visioneer-1, page 11]. Of special note is that Visioneer also allows an end-user to capture web pages, which are also shown as web page visual thumbnails [Visioneer-1, pages 17-19]. Figure VI-8 bottom, which



a) Snapshot of the Visual Explorer, from [Visioneer-1, Page 9]



b) Capturing a web page, from [Visioneer-1, Page 18]

Figure VI-8. Visioneer Visual Explorer [Visioneer-1]

reproduces a portion of page 18 in the user guide, illustrates how end users would do this and what the resulting thumbnail would look like. End users have the option of capturing the portion of it visible on the screen. Alternately, they can direct the system to capture the full page: if the web page is larger than a full screen, it captures it as multiple pages, where it "saves each full screen of information as a separate page in a stack". An end-user can also ask that a previously

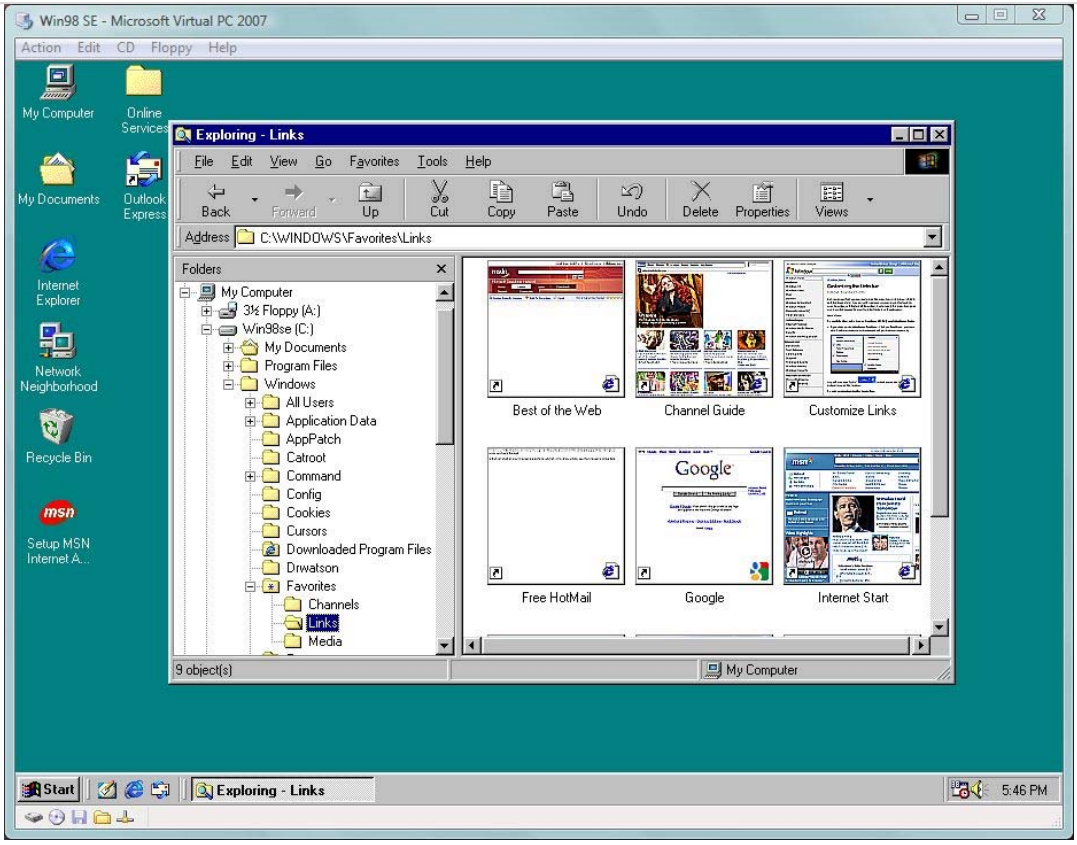


Figure VI-9. Windows Explorer running in a Windows '98 virtual machine showing web page thumbnail

captured web page be updated to show its new content. All these are displayed as thumbnail visual images.

40. Around 1998, Microsoft's Windows Explorer, delivered as part of the Windows 98 environment released on June 1998, included a 'Thumbnail view' option that could be enabled by the end user. This feature, described in [Microsoft-1] and illustrated in Figure VI-9 above is described as follows.

> The image files in the folder are displayed as miniature versions of the actual images in the right pane. The following file types can be displayed by using the Thumbnail view:
> - .bmp (Windows Bitmap files)
> - .gif (Graphics Interchange Format files)
> - .jpg, .jpeg (Joint Photographic Experts Group files)
> -.htm, .html (Hypertext Markup Language files)
> Other file types may be displayed by using the Thumbnail view. [Microsoft-1]

41. Notice that the thumbnails include html files which are web pages. Figure VI-9 illustrates various web page thumbnails as displayed in the Windows Explorer. The folder being viewed is C:\Windows\Favorites\Links. This folder was created and populated by Microsoft's Internet Explorer, where it was used to store links to 'Favorites' web pages (*aka* bookmarks). If the end user had previously selected the 'Enable Thumbnail View', the thumbnail would be stored alongside the link name. My understanding is that this snapshot was produced by running Internet Explorer 4 on a virtual PC environment.

d. Site-Seer: The DOJ v. Microsoft

42. We now turn to systems that showed thumbnail visual images within the context of the Internet, the World Wide Web, and web browsers.

43. In 1998, CNN Interactive provided an example of a page that includes a thumbnail of a home page [CNN-1]. Authors of world wide web pages prior to the '904 patent knew that using visual thumbnails of home pages was beneficial in some cases. One example illustrates this: 'CNN Interactive's Site-Seer: the DOJ v. Microsoft' page dated March 2, 1998. Figure VI-10 below reproduces a portion of this page. As seen, the authors include a thumbnail of the Microsoft home page. Later on in the same page, they also include a home page thumbnail of the US Department of Justice.

44. I have retrieved this page from the internet archive web.archive.org and examined the HTML surrounding the Microsoft Home Page thumbnail, which reads:

<A HREF="http://www.microsoft.com/" target=new><IMG SRC="microsoft.com.jpg" ALT="http://www.microsoft.com/" ALIGN=LEFT WIDTH="199" HEIGHT="208" BORDER="0" HSPACE="10" VSPACE="5"></A>
<P>One of the first sites you might think of visiting is <A HREF="http://www.microsoft.com/" TARGET="NEW">Microsoft</A>. Don't assume you'll be getting an unbiased view in the documents presented in

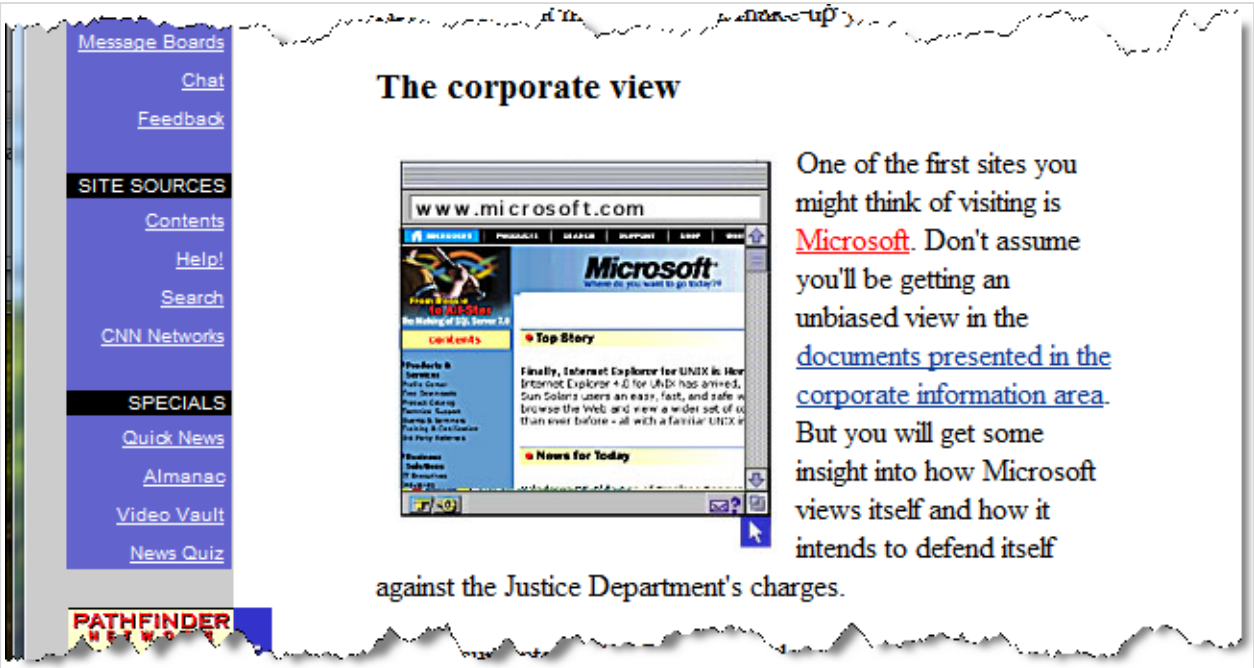


Figure VI-10. Portion of CNN Interactive page Site-Seer: The DOJ v. Microsoft [CNN-1]

the corporate information area. But you will get some insight into how Microsoft views itself and how it intends to defend itself against the Justice Department's charges. </p>

What this says is that the thumbnail image of the Microsoft home page contains a link to the front page of the Microsoft web site http://www.microsoft.com/. If an end user clicked the thumbnail, they would be directed to the Microsoft home page as specified by the microsoft domain name. This HTML also says that:

- the thumbnail image is stored in the file “microsoft.com.jpg”,
- the text http://www.microsoft.com/ should be displayed if the image is not available,
- the ‘Microsoft’ word in the text accompanying and annotating this thumbnail also links to Microsoft front page.
- the phrase ‘documents presented in the corporate information area’ in the text accompanying and annotating this thumbnail links to an interior Microsoft page rather than the front page.

Because the HTML uses the IMG tag to retrieve the Microsoft home page thumbnail image file (IMG is explained in a later section of this report), one skilled in the art would know that this thumbnail image could reside on any server on the Internet that implements the HTTP protocol; that server would then be considered a separate image server from the server serving this particular CNN web page. The code structured around the US Department of Justice home page thumbnail works in a similar fashion.

e. Ayers and Stasko's MosaicG

45. In 1995, Eric Ayers and John Stasko, from the Georgia Institute of Technology, developed the MosaicG system as an improvement upon existing web-based history systems [Ayers-1], i.e., systems that made it easier for a person to navigate to pages they have seen previously. Specifically:

> By presenting titles, Uniform Resource Locators (URLs) and thumbnail images of the documents a user has visited in a session, the Graphic History View allows a user to easily recognize a previously visited document and provides an easy way for the user to re-visit that document and analyze the structure of a set of hypertext documents. [Ayers-1, page 2]

46. MosaicG anticipates the '904 patent in its use of visual thumbnails to represent web pages. In particular, MosaicG presented visited pages as a graphical overview, where one such overview is seen in Figure VI-11 below. As people visited pages, a thumbnail visual image of that page would be captured and the graphical overview updated to reflect that page's position in the web-structure of the hypertext document. A user can select a thumbnail to revisit that page. As Ayers and Stasko describe:

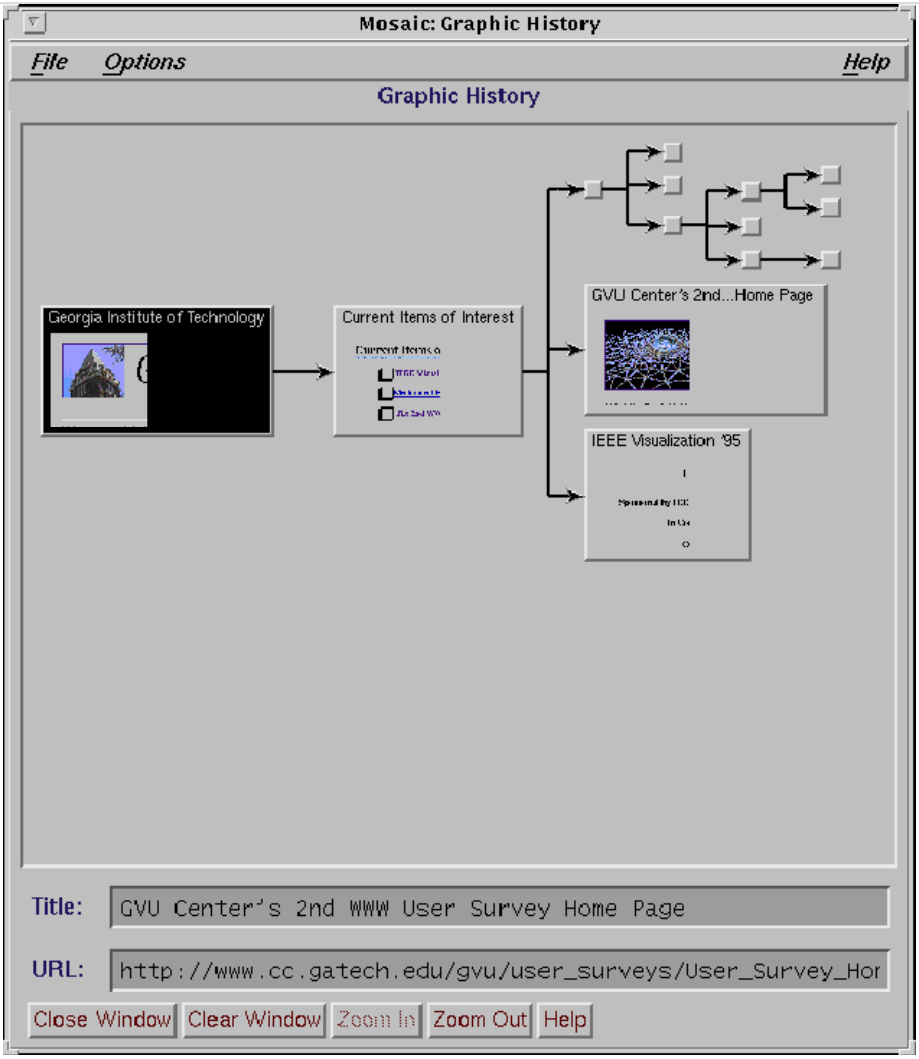


Figure VI-11. MosaicG's Graphical History View, from [Ayers-1, page 5]

> The visual quality of most WWW documents is such that most pages have a distinctive look and feel. MosaicG uses thumbnail images of the documents to allow the user of a browser to quickly recognize a page or set of pages in the tree. A quick glance at the thumbnail representation is often enough for a user to recognize a previously visited page. [Ayers-1, page 4]

Ayer and Stasko also detail a method for rendering thumbnail images [Ayers-1, page 8], where their method anticipates the one described in the '904 patent, i.e., it uses the HTML widget component of a browser to render the page, whose image is then captured and scaled.

47. Ayers and Stasko also recognize that home pages will be captured by this scheme, where they describe the typical pattern of user activity as: "A user visits the document at

the top level, traverses down the tree to read one subject in depth, and then backtracks up to the top node to find another subject." [Ayers-1] Indeed, one such home page is visible at the left side of Figure VI-11, where its title clearly denotes that it is considered a home page by its authors.

48. Ayers and Stasko's MosaicG also anticipates the '904 patent in its provision of a visual image of a home page, where that thumbnail visual image represents links to pages other than the home page. As seen in Figure VI-11 above and described by [Ayers-1], each thumbnail points to links radiating from it, and thus subsumes those links. For example, in Figure VI-11 above, the Home Page on the far left contains links to one child, which in turn contains links to additional children. Each child page, consequently, is visually represented by the site home page thumbnail image. In addition, the user can condense branches of the navigational tree to eliminate visual clutter. Thus the parent node represented by the thumbnail image also acts as a stand-in for the content of these branches and the child pages referred to by it. For example, in Figure VI-11 above the user has collapsed one branch under the 'Current Items of Interest' thumbnail.

49. Ayers and Stasko's MosaicG also suggests the strong relationship between an arbitrary web page and its domain name. As illustrated in Figure VI-11 and described in [Ayers-1], when a user hovers over a thumbnail, the URL appears on the bottom. This is done in left to right order, with the right side being truncated if there is not enough space to fit. That is, the display emphasizes the domain name part of the page path rather than the page name as an additional cue to enhance page recognition.

50. In sum, Ayers and Stasko described a method and implement a system called MosaicG for creating thumbnail images of web sites, used these thumbnail images to

enhance the user experience, and, oftentimes, substituted a "home page" thumbnail image as a representation of children pages. .

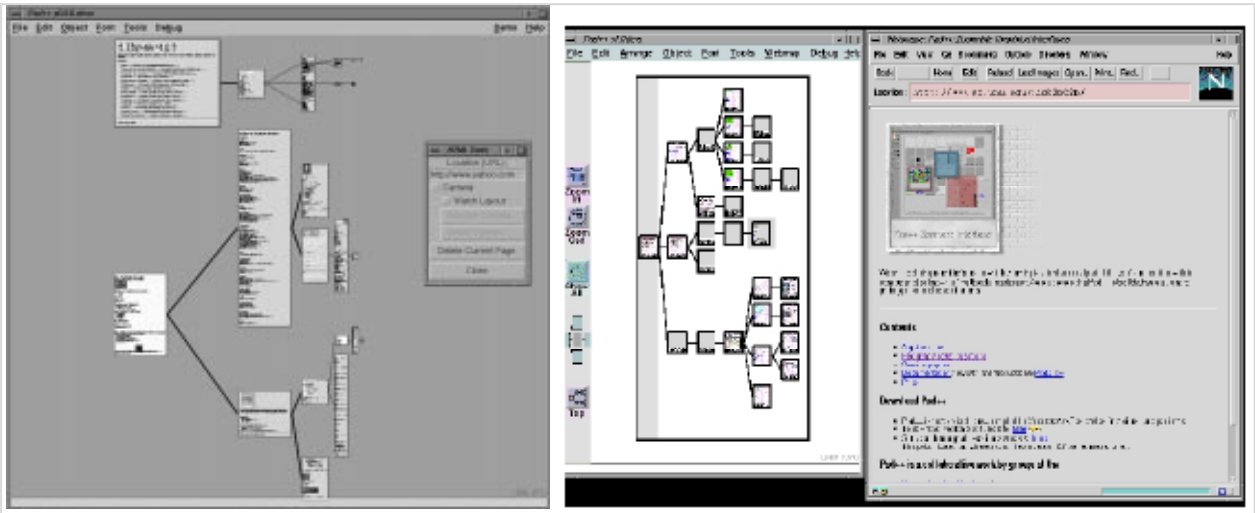
f. Bederson et. al.'s Pad++ Web Browser and PadPrints.

51. In 1996, Ben Bederson and his colleagues from the University of New Mexico described a zooming web browser for navigating the WWW [Bederson-1], with a later version called PadPrints described in [Bederson-2]. [Bederson-3 and 4] are videos illustrating [Bederson-1 and 2] respectively. They describe the system as:

> Instead of having a single page visible at a time, multiple pages and the links between them are depicted on a large zoomable information surface. Pages are scaled so that the page in focus is clearly readable with connected pages shown at smaller scales to provide context. As a link is followed the new page becomes the focus and existing pages are dynamically repositioned and scaled. Layout changes are animated so that the focus page moves smoothly to the center of the display surface while contextual information provided by linked pages scales down. [Bederson-1, page 1 abstract]

52. Their system is based on Pad++, which "allows WWW pages to remain visible at varying scales while they are not specifically being visited, so the viewer may examine many pages at once. In addition, Pad++ allows the user to zoom in and out of pages…". Like MosaicG, when a user clicks on a link to bring up a new page, the page is added to the displayed tree; however, the Pad++ method that determines the tree layout and page size is much more sophisticated. Figure VI-12 below illustrates a snapshot of the Pad++ Web Browser, showing pages of various sizes. Depending on the size, those pages are equivalent to thumbnail visual images.

53. With MosaicG, a user could collapse a sub-hierarchy into a smaller view. Padprints goes one step further, where "the multiscale facilities of Pad++ allow natural ways to temporarily remove sub-hierarchies from view by shrinking them until they occupy minimal



(a) from [Bederson-1] **(b) from [Bederson-2]**

Figure VI-12. Pad++ Web Browser and PadPrints, from [Bederson-1]

screen area." Again, this means that the parent thumbnail represents multiple links to its children, rather than a single link to its page.

54. They also describe their system as a browser companion [Bederson-2], which anticipates the '904 description of thumbnail visual images being displayed alongside the visual image of the web page. This is illustrated in Figure VI-12 right, where the thumbnail tree is shown on the left side of the display and the full size browser on the right side.

g. Cockburn, Greenberg, Kaasten et. al.'s WebView and Grouplab History.

55. As mentioned in the background, I was directly involved in the development of history systems that used thumbnail visual images to portray web pages to a person. This began with my work with student Tauscher, published between 1996 and 1997 [Tauscher-1,2 & 3], where we offered a guideline recommending that interface designers should, when presenting a list of web items to the user, use a meaningful representation of a web page. One of the suggested representations is small visual thumbnails of the web page.

56. In 1999, I and my collaborators, including Andy Cockburn, described variations and intellectual foundations of the WebView system [Cockburn-1,2 & 3]. As with MosaicG and PadPrints, WebView also presented web pages as a collapsible structural tree of visual thumbnails, except this time organized under its common root in the domain path *vs.* the visitation sequence (see Figure II-2, left side, taken from [Cockburn-1]). That is, the thumbnail representing a collapsed home page (typically the left-most page displayed in the hierarchy, as in Figure II-2 left) represented a multitude of its children links. Importantly, each thumbnail visual image also contained a list of all links within it, which was visualized as a popup menu when the end-user right-clicked the thumbnail (see Figure II-2, right side, taken from [Cockburn-1]. If the thumbnail was a home page, it contained a link not only to itself but to other pages that shared the domain name as its common prefix. Both these points directly anticipate the use of a home page thumbnail visual image to represent pages other than itself.

57. In 1999, I and my student Kaasten designed and implemented another history system, illustrated in Figure II-1 and first published in 1999 [Kaasten-1], which used semantically enhanced thumbnail visual images. That is, thumbnails were modified to show the end-user's frequency of visitation to the page it represented, as well as whether it was a bookmark. As part of the same system, we knew that some pages were best organized by domain name. Thus we added a drop-down menu (the 'Pages From: menu at the top left of Figure II-1) that parsed all the visited pages for their domain names, and then presented those domains in the menu. When a domain name was selected, only those thumbnails representing pages in that domain were shown in the visual list. The presentation of the domain name at the top of the pull-down menu was accompanied by the visual thumbnail of that domain name. This feature was demonstrated at the 1999 HCIL workshop [Kaasten-1]; our video [Kaasten-3] and Figure 2 from

[Kaasten-3] (reproduced below) illustrates the domain name thumbnail as shown in the demonstration).

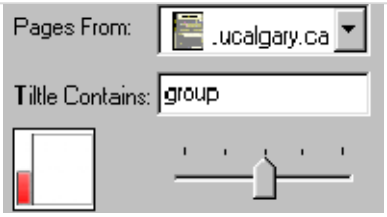


Figure VI-13. The domain-name thumbnail from [Kaasten-3-1]

58. In both the above systems, the thumbnails were displayed in a window alongside the browser containing the main page. In Webview, this was a detached window (seen in Figure II-2), while in the Kaasten system, it was a sidebar integrated with the Internet Explorer web browser (as seen in Figure II-1). Both systems included a popup display of thumbnail visual images as well. If a user hovered the mouse over the thumbnail, a larger image of the same web page would appear atop of it to provide the end-user with more information – and better recognizability – of the web page it represented.

h. Robertson et. al's Data Mountain

59. In 1998, Robertson et. al. from Microsoft Research [Robertson-1, Robertson-2] described and presented the Data Mountain:

> …a novel user interface for document management designed specifically to take advantage of human spatial memory (i.e., the ability to remember where you put something). […The] user freely arranges document thumbnails on an inclined plane textured with passive landmarks. [Robertson-1]

The documents described in this passage are web pages, and the thumbnails are visual images of these web pages. People can move these thumbnails around the plane, and the particular size of the thumbnail depends on its position on the plane. When the user selects a web page, it appears

Figure VI-14. The Data Mountain, from [Robertson-1]

in readable size at the front of the plane. Aside from the novelty of the visualization, the primary idea is that people can more easily recognize and recall particular pages due to not only the thumbnail image but from spatial position as well, i.e., where they rely on spatial memory. Figure VI-14 above taken from [Roberston-1] illustrates the various thumbnails laid out on this inclined plane, as well as the selected page viewable at the front. [Robertson-2] is a video illustrating features of DataMountain. As well, it shows a user interacting with a home page thumbnail visual image placed on the DataMountain, where manipulating it brings the full size home page into view at time 1:34.

i. Card, et. al.'s Web Book / Web Forager

60. The systems described from sub-sections e through h mostly concern themselves with capturing and recording a historical record of those pages that an end-user had previously seen or visited, where pages are then presented to the end-user as thumbnail visual images for later recall. However, there were a variety of other applications for thumbnail visual images within the context of the Internet and the World Wide Web, where thumbnails are used to

primarily to illustrate collections of pages that an end-user may or may not have seen before. These are presented next.

61. In 1996, Card, Robertson and York, working at Xerox PARC, developed the WebBook and Web Forager system. The WebBook is a 3D interactive book of HTML pages, while the Web Forager embeds the WebBook and other objects into a hierarchical 3D information workspace. [Card-1] describes the system as well as its intellectual foundation, while [Card-2] is a video and accompanying abstract illustrating system features. Robertson et. al. were concerned that end users using contemporary web browsers saw only one page at a time, with no difference between a page "in the collection" (i.e., relative URLs collected under a typical home page) and one outside the collection. As an alternative, they proposed the WebBook to represent such a collection: "Given a collection of web pages, [the WebBook] preloads those pages and displays them as a collection using an augmented simulation of a physical book [Card-1, page 113]". Figure VI-15 below reproduces screen snapshots of WebBook from [Card-1], where progressive screens show the book opened up on page, interactively ruffling pages in the book, an alternate view of the book through a 'Document Lens', and the book as seen in relation to other books in a 3D space. Depending on the size of the book and its pages as presented in the 3D space (as illustrated in Figure VI-15(d)), the rendered image can be considered by the end-user as representing the page or as a visual thumbnail.

(a) WebBook

(b) ruffling pages in WebBook

(c) WebBook viewed with a Document Lens

(d) Web Forager

Figure VI-15. The Web Book, reproducing Figures 2,3,4 and 5 from [Card-1]

62. The thumbnail visual image also contains multiple links to pages other than itself. As Card, et. al. write of each visual image:

> Links are color coded so the user can easily tell the difference between a reference to another page in the book (red links) and a reference outside the book (blue links). Picking a red link will animate the flipping of pages to the desired page. Picking a blue link will close the current WebBook and look for the page elsewhere. If the page is in another WebBook stored on a bookshelf, that WebBook is opened to the desired page. If the page is in none of the WebBooks. then the Web Forager is used to display the individual page in the user's information workspace. [Card-1, page 113]

63. The Document Lens view of a book shows a whole collection of pages at a glance: pages are displayed as selectable thumbnail visual images as illustrated in Figure VI-15(d). Card et. al. explain this, where Fig. 4 is the one shown in Figure VI-15(d) above:

> Books are compact but (except for bookmarks) sequential. Therefore we allow the user to explode the book out (in animation) so that all the pages are available simultaneously. The Document Lens [19] can then be used to inspect portions of interest. Fig. 4 shows the WebBook Document Lens view. The user is then able to pan and zoom over the entire set of pages, while retaining a focus plus context display of the book. When returning to the book view, the user sees an animation of the pages imploding into a book. [Card-1, page 113]

64. Card et. al. also explain various methods to collect the URLs of web pages to include in the book. These include *home page books*, which includes those pages referred to directly by a given page, *topic books* that collect sets of pages related to a topic, *search reports* that "can be used to display the results of a content-based search, either keyword based or relevance based". As seen in Figure VI-15(d) such books are represented by either a title page or a representative page, which depending on their size could be seen as a thumbnail visual image. Thus the thumbnail visual image of the book's cover or front page represents pages other than the one displayed by that page. In the case of home page books, this would include the thumbnail of the home page representing pages linked to by that home page.

j. Wittenburg et. al.'s Polynav

65. Also in 1998, Wittenburg et. al. from GTE Laboratories [Wittenburg-1] focused on allowing end-users to rapidly preview a collection of web pages by seeing a sequence of thumbnail images culled from those pages. These web page collections and derived sequences can come from a variety of sources, including search results or out-going links from a given web page [Wittenburg-1, page 79]:

> (1) links out (with specified depth limit) from one or more start URLs
> (2) ranges of results from a query (first 30, second 30...)
> (3) different sortings of a page set (e.g., by name, by last-modified date, by frequency of visit, by profile match, etc.)

Figure VI-16. The VRML version of PolyNav, from [Wittenburg-1]

66. Their system, called Polynav, presents a visualization – a stream of thumbnail images – that allows end-users to rapidly preview images (out-links) culled from the pages in a given collection. Depending on the Polynav version, end users can 'play' those images as a sequence, where new images are placed atop old ones to create a layered collage or by having images slide across a viewing axis. Figure VI-16 above, taken from [Wittenburg-1, Figure 2], shows one version of Polynav, where the end-user is previewing the contents of a page set representing the 'Shop Online Service' of a web site. What is not evident in the figure is that the image thumbnails are animated, where they are displayed as a moving sequence.

k. Mandala and the Imago Image Server by Helfman

67. In June of 1999, Jonathan Helfman from AT&T Labs and the University of New Mexico, USA presented Mandala, yet another system that uses thumbnail images to represent, access and organize web information (such as a user's browsing history results or

results of a query as previously discussed) in a way that makes them comprehensible to end users [Helfman-1]. As Helfman writes:

> Mandala lets people visualize large groups of web pages by displaying selectable thumbnails of the pages' images. Groups may be determined in many ways: the URLs in a bookmark file, the history of a browsing session, the results of a query, etc. Mandala's displays function as visual interactive indexes; they provide an overview of large amounts of information without sacrificing access. [Helfman-1, page.164]

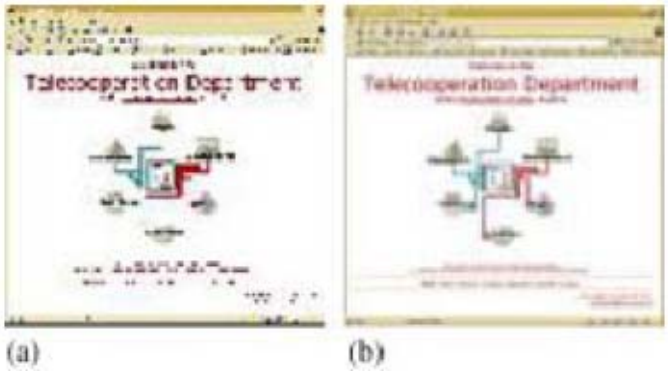
Helfman also discloses that these images have underlying links: "Double-clicking on an image causes the client to signal the Mandala server, which signals a web browser (via its client API) to display the associated page [p168]". Furthermore, in discussing their image server, Helfman describes how Mandala's image server, called Imago, creates thumbnails from URLs supplied to it:

> Mandala's image server, called Imago, has been developed to support fast image data compression and decompression, image shrinking, imagemap creation, and image meta-data extraction (i.e. information about the image that is normally hidden in the image's header, such as its dimensions or total number of colors). … Imago creates thumbnails and imagemaps according to client specifications. A minimal thumbnail specification is a URL for the input image, which causes Imago to use default scale factors and filtering functions. Clients can specify the maximum thumbnail dimension, filter function, return style (whether to return the URL or the data), and replacement style (whether to overwrite an existing thumbnail of the same name or generate a new name. [Helfman-1, pp165-166]

I will provide further details of Mandala's image servers in a later section of this report.

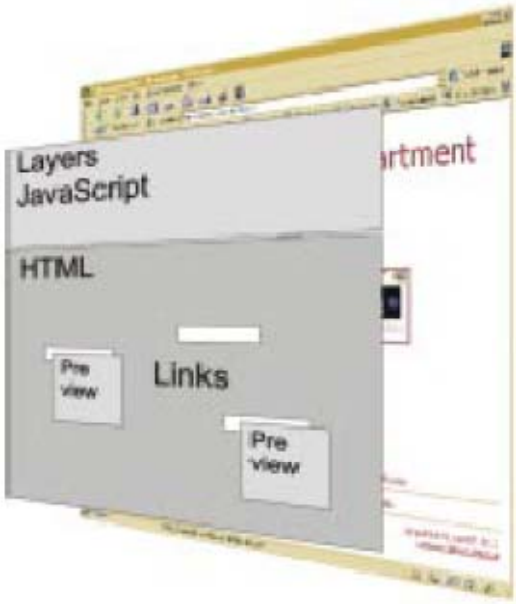
l. Kopetzky et. al.'s Visual Link Previews

68. In 1999, Kopetzky and Muhlhauser from Austria [Kopetzky-1] described and built a system that creates visual previews of links on a web page for view by end users for the purpose of helping them make a "decision about which links to follow and which to ignore" [Kopetzky-1, page 1525]. While they describe various previewing methods, they focus on thumbnail visual images of web pages, which they call thumbnail previews. As part of their



[Kopetsky-1, Figure 2]
a) pixel resizing… b) bilinear resampling…

[Kopetsky-1, Figure 5] *The mouse pointer (not shown) moved over the Blackboard symbol and the preview opened*



[Kopetsky-1, Figure 7] *Basic architecture of the proxy server*

[Kopetsky-1, Figure 9] *Client side architecture*

Figure VI-17. Visual Link Previews, from [Kopetzky-1]

discussion, they describe properties of these thumbnail previews, e.g., that the image should be small, in color, and that "the quality of the image should be good enough to allow the reader to recall the accompanying image of the target of the link from memory" [Kopetzky-1, page 1527]. Figure VI-17 above reproduces: Figure 2 (top left), Figure 5 (top right), Figure 7 (bottom left) and Figure 9 (bottom right), all from [Kopetzky-1]. The top left figure shows two thumbnails of a web page, showing how the image quality is affected by the sampling algorithm used to create

the thumbnail. The top right shows what the preview looks like within the client-side browser after the user moves their cursor over a link (only a portion of the browser window is shown). The bottom left shows how the Netscape browser client implements these previews as a layer, although Kopetzky said this could also have been done in Internet Explorer using cascading style sheets. The bottom right shows how the proxy server is separate from the browser client and from the parser that analyzes the document's structure and searches it for link information. Kopetzky et. al. also discloses displaying the thumbnail appearing both within the visual image of the page, and hovering above it.

69. Kopetzky et. al. also provides an example of a thumbnail preview of a home page in his discussion of link types. Note that the Figure 2b he is referring to is the thumbnail image shown at the top right of Figure VI-17 above. Note too that he describes how the URL can be examined (parsed) to discover what kind of link it is:

> Both links have explicit types, which can be used for preview purposes. The following list shows which link types are recognized by our system and how they are visualized. **As the linking mechanism works with URLs we are using properties of the URLs to categorize a link.**
> * **The URL points to the beginning of a Web page, as in http://www.tk.uni-linz.ac.at/. Links of this type are visualized using a thumbnail picture as in Fig. 2b.**
> * The URL points to an anchor, as in http://www.encyclopedia.com/h.html#hypertext. Links of this type may be visualized using a thumbnail picture or, if there is text after the anchor, the text referenced by the link itself may be displayed (see Fig. 3a).
> *The URL starts with a protocol different from HTTP, like FTP, TELNET, MAILTO, etc. Although these link types are rather self-explaining a symbol is used to represent the protocol used (see Fig. 3b for an example).
> In addition to these properties links can have meta-properties, like *dead* or *forbidden.* … The symbol chosen for a dead link is presented in Fig. 4. [Kopetzky-1, pages 1527-28, bold emphasis added]

Kopetzky et. al. also disclose a separate image server for rending and storing thumbnail visual images; this will be discussed in a later section.

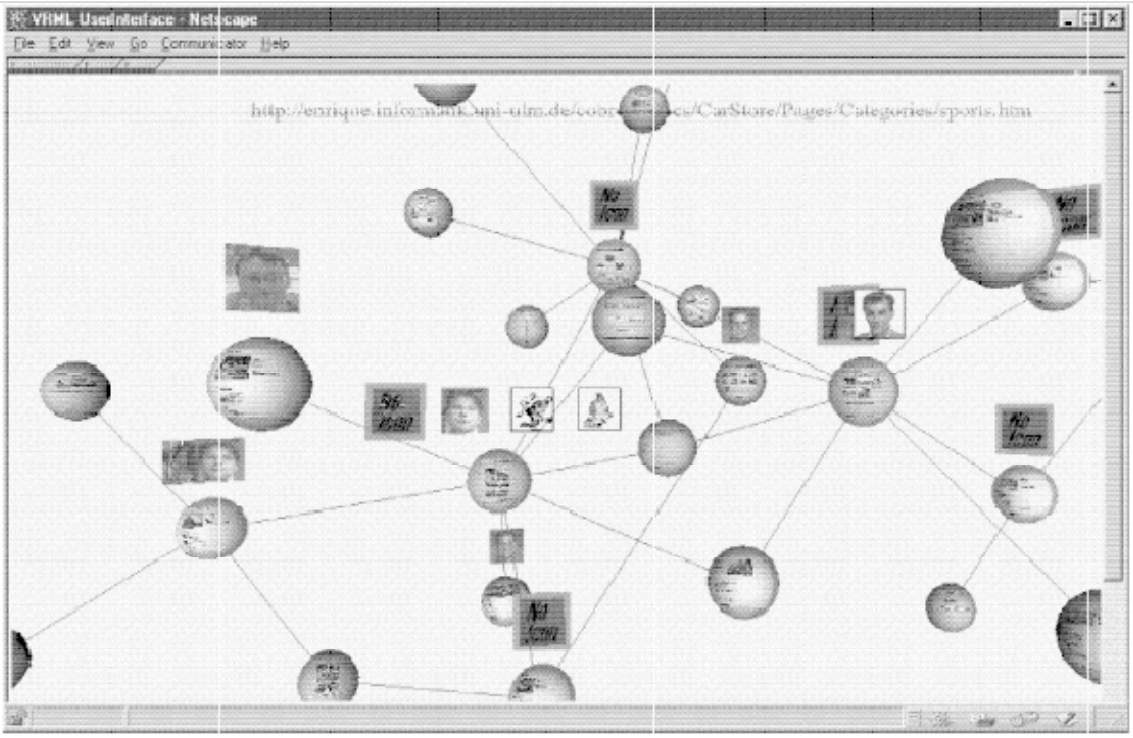


Figure VI-18. CoBrow VRML User Interface, from [Schmid-1]

m. Schmid et. al.'s CoBrow

70. In 1998, Schmid et. al. from the University of Ulm, Germany [Schmid-1] described and built a system that visualizes a user's vicinity – "the user's neighbors and the Web pages closely linked to the page the user reads" – while browsing the web. By neighbors, he meant "people who read the same Web page or browse the same area of the web". As with the many other people working in this area, he too recognized the importance of thumbnail visual images to, in his own words "improve the power of expression of hyper-links" [Schmindt-1, page 1]. They then developed the CoBrow system to visualize such a neighborhood as a graph that combines both thumbnail visual images and user icons representing the neighbors. Figure VI-18 above reproduces Figure 3 from [Schmid-1], where it shows a screen snapshot of the system in use: both web page thumbnails and user icons are included in this snapshot. Schmid et.

al. goes further, where they also describe the underlying system and architecture. They distinguish between *static thumbnails* that are pre-rendered and reduced-size images, *vs.* dynamic thumbnails that are "online provided and on demand rendered images of Web pages." As we will see later, these on-demand images are provided by an image server.

n. The Auditorium seating visualization, by Terveen et. al.

71. In 1998, Terveen et. al. from AT&T Labs [Terveen-1, Terveen-2] presented another approach and system that constructs, organizes, and visualizes collections of topically related web pages in a way that makes them comprehensible to end users. In their own words:

> we have created a novel visualization, the auditorium view. This is an exploratory interface that lets users quickly identify sites by a graphical thumbnail image, pick out sites that are most central to the topic (based on interconnectivity with other sites), arrange sites by any of the properties in the site profiles, and to explore the profile information for sites and their constituent pages. [Terveen-1]

72. A key claim in [Terveen-1] is that "the Web site is a more appropriate basic unit than the Web page". For each site, they then generate a site profile, which includes amongst other things a crawler to gather home page thumbnail images:

> The crawler analyzes the content of pages it fetches in order to build profiles of the site content and structure. Profiles include the following data:
> -Title (of the site's root page);
> - A thumbnail image (of the site's root page)'
> - Links to and from other sites'
> - Media contents of pages and sites, including images, audio files, and movie files;
> - Internal pages of the site... we can provide access to these internal pages in the interface, thus offering "shortcuts" to interesting content…"
> [Terveen-3, page 15; also Terveen-1, page 83]

73. [Terveen-1] described a method where they calculate the site root page by looking for a common root between multiple pages. In their other papers they also present

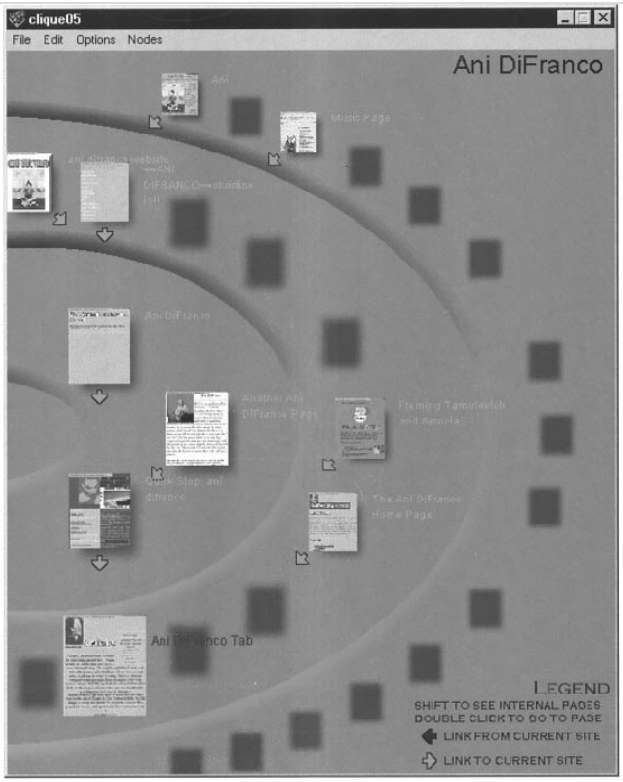


Figure VI-19. The Auditorium view of the Clan Graph, from [Terveen-1]

various heuristics for determining if internal pages also represent distinct sites. In [Terveen-3], they offer the following heuristic:

> "General heuristics include the convention that the tilde (~) character indicates different user sites (therefore, http://www.research.att.com/~terveen and http://www.research.att.com/~willhill are usually distinct sites). Site-specific heuristics encode knowledge that enables the identification of distinct user sites, hosted on large servers such as GeoCities or Tripod. Of course, the site aggregation rule and our heuristics can fail; therefore, we also have explored heuristics to detect when sites should be split or merged." [Terveen-3, pp. 14-15]

74. Terveen et. al. then created a visualization called the auditorium view that "gives a graphical overview of the most important several dozen sites for a topic, [and] lets users explore structural relationships between sites and the internal structure of individual sites…" [Terveen-1, page 92]. Figure VI-19 above, which reproduces Figure 5 from [Terveen-1] is snapshot of this system. As seen in the image and explained by Terveen et. al., the Auditorium View presents (amongst other things) "thumbnails of site 'front door' pages [that] serve as iconic

representations of sites." This view is linked to a web browser, where clicking on a thumbnail drives the browser to that web site. However, other information is associated with that thumbnail. As Terveen explains [Terveen-1, page 90]: "Users move the mouse cursor over a site to focus on it, and only links from or to the focused site are shown. To further reduce clutter, we do not draw complete links between sites, since they draw too much user attention to uninformative crossings and edge angles. Instead, we represent links with small in and out arrows. … In addition, by clicking on a site, the user can access a display of the site profile data to find the amount and type of information the site contains and access significant internal pages."

o. US Patent 6,356,908 (Brown et. al.)

75. On July 30 1999, Brown et. al. (IBM) filed a patent titled *Automatic web page thumbnail generation* [US Patent 6,356,908][2]. Although Brown was of record in the file history, I note that the Examiner stated that he was not relying on this particular patent, and instead discussed a different patent by Brown at length in the patent prosecution. Brown begins by identifying the problem he wishes to solve. He describes how search engines and other web pages typically "return a list of links of relevant sites with perhaps a short verbal description of the site" [US Patent 6,356,908, col. 1:52], which he then claims "does not provide sufficient information to enable one to make an intelligent decision as to whether to follow the link" [US Patent 6,356,908, col. 1:57]. He then describes an invention and method that presents such content by thumbnail visual images, where his patent includes a method for generating and storing these thumbnails. Figure VI-20 below reproduced Figures 8, 12 and 14 from US Patent

[2] The '904 Patent cites two patents by Brown: 6,356,908 (the "'908 Patent") and 6,665,838 (the "'838 Patent"). Only the '838 Patent is discussed in the '904 file history. The '908 Patent, charted here, was "made of record and not relied upon" but was "considered pertinent to applicant's disclosure." '904 Patent file history, Office Action dated March 8, 2004, page 11.

6,356,908. The first two figures illustrate how his method embeds the generated thumbnail visual images within the web page seen by the end-user (in this case showing search results). US Patent 6,356,908 Figure 8 (shown in Figure VI-20, top) clearly shows thumbnail visual images within the visual image of the web page. US Patent 6,356,908 Figure 12 (shown in Figure VI-20, middle) shows thumbnail visual images hovering over a hyperlink as the user passes the pointer over it. US Patent 6,356,908 Figure 14 (shown in Figure VI-20, bottom) shows the thumbnails of links displayed in a separate window.

76. Importantly, Brown also anticipates 'providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink'. He teaches, in one of his embodiments, that the displayed thumbnail can be an iconic representation of the domain rather than the visual image of the actual page specified by the link.

> In another embodiment of the pop-up thumbnails, **rather than generating and displaying thumbnails of the web pages associated with links, an icon representing the domain of that link could be generated and displayed next to the text representing the link.** For example, if the domain is associated with Yahoo, then an icon displaying the Yahoo logo might be displayed next to the link. Furthermore, the icon could be assigned by the user or by the domain itself, and picked up automatically by the browser. This icon would then pop-up next to the link as the pointer moves over the link, just as the thumbnail does in an example depicted in FIG. 12." (emphasis added). [US Patent 6,356,908, col. 8:16-27]

Brown generalizes his idea to icons. However, using a thumbnail image as an icon was known in the art. For example, as Cok in US Patent 5,548,692 (filed in 1995) explains:

> "Icons representing an object within a GUI are generally symbolic, pictorial rectangles of a common size. This facilitates the effective use of space and user recognition of the objects represented. . . . When icons represent image objects within a GUI, it is helpful to use a very small version of the image itself as an icon rather than a standardized image of a symbol. These small image representations are also known as thumbnail images. The thumbnail image clearly represents the image content since it is simply a miniature version of the image." [US Patent 5,548,692, col. 1:41-56]

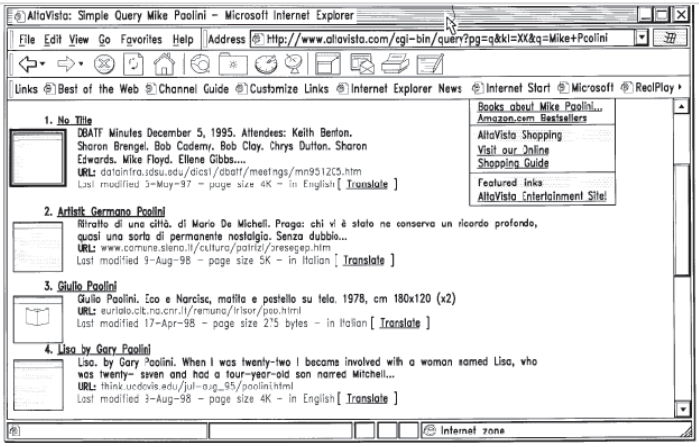


Figure 8, from [US Patent 6,356,908]

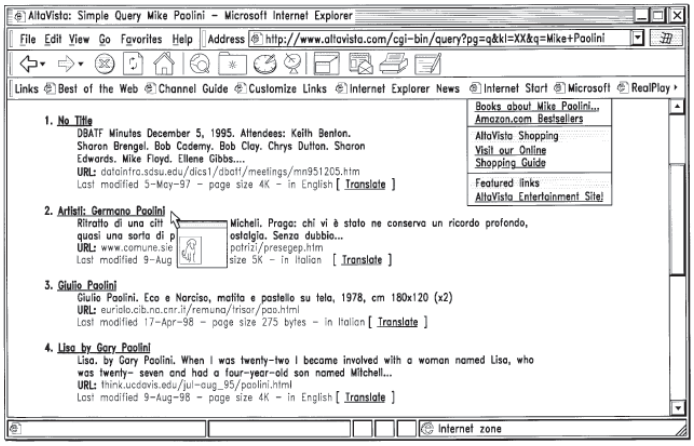


Figure 12, from [US Patent 6,356,908]

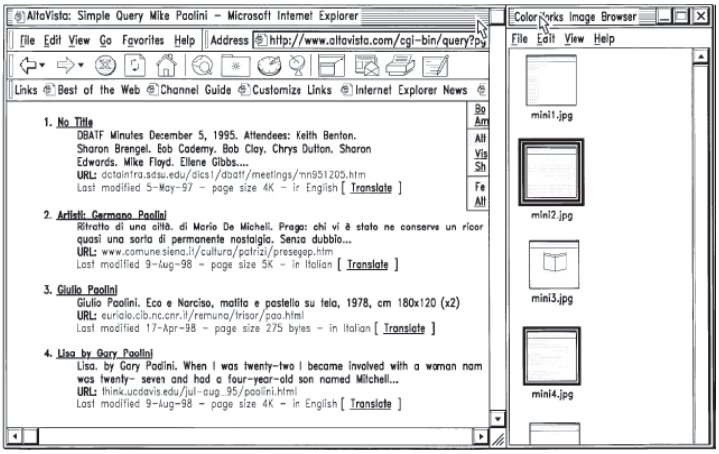


Figure 14, from [US Patent 6,356,908]

Figure VI-20: Figure extracts from [US Patent 6,356,908]

Furthermore, thumbnails fit the definition of an “icon” as defined in semiotics and visual interface design, where thumbnails fit the definition of highly concrete iconic representations. As discussed in Mullet & Sano’s 1995 book ‘Designing Visual Interfaces’:

> "The critical process of representation (the focus of analysis at the *semantic* level) depends on establishing a clear relationship between a representamen and its object. Pierce (1931) identifies three forms this relationship can take. **An *icon* (184-a, 185-a) denotes its object by virtue of its own likeness to or resemblance of that object, on the basis of some quality or characteristic inherent in the icon itself.**" [Mullet & Sano, pp. 172-73, emphasis added]

> "Finally, a given sign representation can be characterized by its degree of abstraction – the extent to which the essential qualities upon which the representation is based are isolated from the literal perceptual characteristics of the sign object. **A photograph or realistic illustration provides a high degree of fidelity to a particular sign object (an instance of its class), and is usually easy to recognize as a result**." [Mullet & Sano, p. 174, emphasis added]

77. Brown also anticipates 'providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink'. In particular, Brown teaches that thumbnails can be modified, where the modified thumbnail is displayed instead of the original thumbnail (e.g., see steps 765 and 770 in the '908 patent). He describes an example where a filtering program has detected undesirable content on a linked page in a search result, which then modifies the thumbnail to portray a diamond with a 'do not enter' message. He then says that other modifications to the thumbnail images will be obvious to those skilled in the art.

p. US Patent 7,177,948 (Kraft et. al.)

78. On November 18, 1999, Kraft et. al. (IBM) filed a patent titled *Method and apparatus for enhancing online searching* [US Patent 7,177,948]. As with Brown [US Patent 6,356,908], Kraft argues that search result page as displayed in a browser should be augmented by visual abstracts (thumbnail visual images), and adds that these thumbnails can be of different sizes. Figure VI-21 below reproduces Figures 1 and 2 from this patent. The left figure illustrates how:

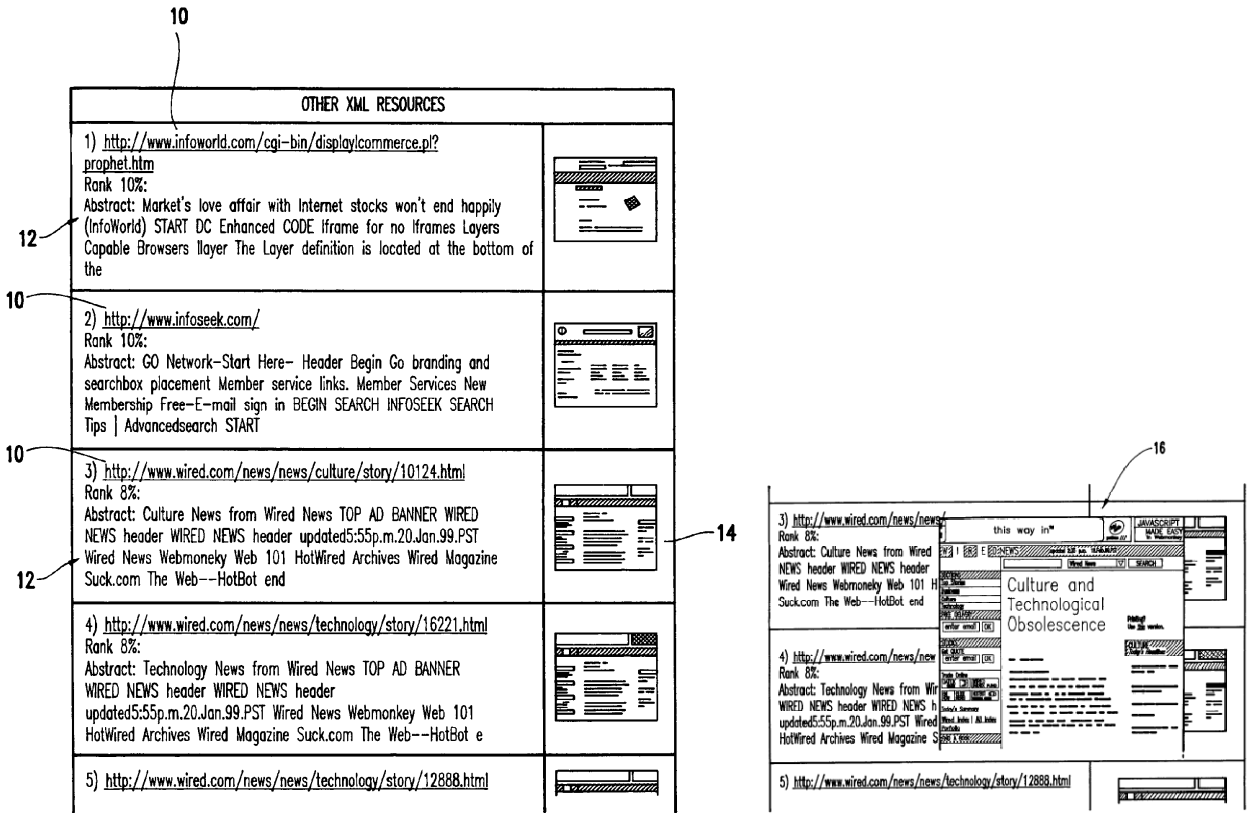


Figure VI-21. Figures 1 (left) and 2 (right) from [US Patent 7,177,948]]

> The result page 5 generally includes a short summary description 12 and a visual abstract (i.e., thumbnail) image 14 for each document found in the search. [US Patent 7,177,948, col. 4:63-5:1]

These figures also illustrate how his method embeds the generated thumbnail visual images within the web page seen by the end-user. The left figure clearly shows thumbnail visual images within the visual image of the web page, while the right figure shows the larger thumbnail visual image hovering over a hyperlink "when a cursor is moved over the smaller one of the visual abstracts" [US Patent 7,177,948, col. 3:12]. Additionally, Figure 1 (as shown in Figure VI-21, left) shows an embedded thumbnail image of the home page of the "www.infoseek.com" website. As with Brown [US Patent 6,356,908], Kraft also offers a method for generating these thumbnails, and for using image servers; these will be discussed later.

q. US Patent 6,058,417 (Hess)

79. The previous examples show that the use of thumbnails was a well-known method for showing scaled visual images of web pages as a way to enhance the meaning and recognizability of a link. What they don't show is that similar uses of thumbnails (as well as image servers, discussed later) were widely known across other Internet applications, i.e., it was well known general technique rather than one restricted only to web pages. On November October 13, 1997, Hess et. al. (eBay) filed a patent titled *Information Presentation and Management in an Online Trading Environment* [US Patent 6,058,417]. Hess's patent is directed for online trading environments, where:

> Images are harvested from a plurality of sites based on user-supplied information. The user-supplied information includes descriptions of items for sale and locations from which images that are to be associated with the items can be retrieved. Thumbnail images are created corresponding to the harvested images and are aggregated onto a web page for presentation at a remote site." [US Patent 6,058,417, col. 2:13-21]

Hess clearly states that the ideas in his patent were generalizable to other applications that collected image data and presented it to a user; web page image thumbnails falls under this description, as do web pages that consist solely of an image file:

> Importantly, while embodiments of the present invention will be described with respect to an online person-to-person trading environment, the method and apparatus described herein are equally relevant to other applications in which image data is collected from disparate sources and presented to a user . . ." [US Patent 6,058,417, col. 3:65-4:4]

r. Summary

80. The above background is an illustrative rather than exhaustive overview of how thumbnail visual images have been developed and used by many others in ways that

anticipate various purportedly inventive elements of the ‘904 patent. We have seen that thumbnail visual images have been used regularly to:

- represent documents, images, web pages, and other file types in desktop browsers and specialized applications,
- within a web page, where the page author used a web page thumbnail to represent another web page (including home pages),
- within various systems to visually represent web pages that the end user had previously visited (including home pages),
- within web browsers and systems that collect web page collections, e.g., search engine results.

We have also seen that thumbnail visual images can be of home pages, where the thumbnail can represent:

- the home page itself and a link to itself,
- the home page alongside a link to other pages,
- the home page, which collects, displays, and allows an end user to select from a plurality of links that it represents,
- the home page, where its thumbnail acts as a hierarchical entry point to an expandable list of various children thumbnails.

We have also seen that these thumbnails are presented in many ways that again anticipate various elements of the ‘904 patent:

- as one or more thumbnails presented within the visual image of a web page,
- as thumbnails appearing hovering over a hyperlink in a web page,
- as thumbnails appearing alongside the visual image of a web page.

B. **IMAGE SERVERS**

81. Image servers were a very well known concept prior to the ‘904 patent. They were known in non-Internet domains, such as in storing and serving medical images, e.g., the previously discussed Image Engine in [Lowe-1] which also served visual thumbnails of these images. For brevity, I will focus this history on image servers used in hypertext generally, and specifically in the Internet.

a. Vannevar Bush and Memex

82. I already introduced Bush’s seminal *memex* system that he had envisioned in 1945 [Bush-1]. Recall his description: “a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility.”, how it stored images 20 – 100x smaller than the original, how he envisioned “a library of a million volumes could be compressed into one end of a desk”, and how it could be presumably operated from a distance. Thus Bush foreshadowed and anticipated the image server described in the ‘904 patent 55 years before the patent filing date. The microfilm record is a true image server, as it photographically stores all its contents as microfilm images and serves these images to the user via the *memex* device. It is also a thumbnail image server, as these images are between 20-100x smaller than the original. That he saw this as a separate image server is clear from his statement that it could be operated at a distance.

b. The HTML <IMG> Tag

83. A fundamental aspect of HTML at the time of the patent was the <IMG> tag, which specifies that an image should be included in the document that will be eventually rendered on the end-user’s display. What is important is that the image can be stored anywhere

on the Internet, perhaps on the local computer, or perhaps on a different separate computer that handles HTTP requests. This means that the computer serving the web page can indicate that an image should be retrieved from another separate computer. By definition, this separate computer behaves as an image server separate from the original web server because it can respond to this request from any client. As well, a link associated with the image could be any hyperlink.

84. The IMG idea precedes the '904 patent by many years, was a fundamental part of the way the web worked. On February 25, 1993 Mark Andreesen proposed the IMG tag specification [Andreesen-1], where he provided an example of the IMG format: <IMG SRC="file://foobar.com/foo/bar/blargh.xbm>. Andreesen stated that the argument - the SRC and path after IMG - behaves as follows: "This names a bitmap or pixmap file for the browser to attempt to pull over the network and interpret as an image, to be embedded in the text at the point of the tag's occurrence." This proposal was included up as part of the HTML 2.0 standard in 1995, where it appears in the Hypertext Markup Language – 2.0 specification [RFC1866]. Quoting Section 5.10 (where a URI is a Uniform Resource Identifier that includes URLs):

> 5.10. Image: IMG
> The <IMG> element refers to an image or icon via a hyperlink (see 7.3, "Simultaneous Presentation of Image Resources").
> ...
> SRC specifies the URI of the image resource.
> ...
> Examples of use:
> <IMG SRC="triangle.xbm" ALT="Warning:"> Be sure to read these instructions.
> <IMG SRC="sample.xbm" ISMAP> "

The standard also anticipates 'a multiplicity of downloaders' in Section 7.3, where "HTML user agent may activate hyperlinks indicated by <IMG> and <INPUT> elements concurrently with processing the document; that is, image hyperlinks may be processed without explicit request by

the user." What this meant is that multiple image requests could be sent to one or more servers at the same time, which meant they were handled partially concurrently rather than sequentially.

c. The Common Gateway Interface (CGI)

85. The idea of separate servers is a well known and fundamental aspect of the Internet. Well before the '904 patent (circa 1993 to 1995), the Common Gateway Interface (CGI) standard was developed to define the protocol for how an external application (such as a web browser) could talk to an information server on the internet (such as a specialized web server). Typical, the client would make a request via CGI. The server would process the request in real time, and generate output that would be sent back to the client. In 1995, David Robinson described version 1.1 of CGI as follows:

> The Common Gateway Interface (CGI) is a simple interface for running external programs, software or gateways under an information server, in a platform-independent manner. Currently, the supported information servers are HTTP servers. The interface has been in use by the WWW since 1993. [Robinson-1, Abstract]
>
> Together the HTTP server (httpd) and the CGI programs are responsible for servicing a client request by sending back responses. The client request comprises a Universal Resource Identifier (URI) [1], a request method, and various ancillary information about the request provided by the transport mechanism. [Robinson-1, Introduction]

86. A common use of CGI was as a gateway to a database. For example, if one wanted to connect a database to the world wide web, one would create a server that receives the CGI request from a client, transmits that request to the database engine in a form understandable to it, receive the results from that database request, and send those results back to the client in an understandable form. An image server using an image database or other means of accessing and/or storing images is well within the scope of CGI, and would be known to one skilled in the art.

d. Greenberg's Assignment 3

87. In 1996, I gave my students in my CPSC 441 class an assignment titled 'Building An Image Web Crawler and a Web Server.' The course assignment as given out is in [Greenberg-3]. The assignment contained 3 parts. The first part asked students to build a program and web crawler that downloaded images from the web into a file store. The second part asked them to create a web server – specifically an image server – that returned either a specific named image or a page containing all images it had stored. The following extract from the assignment gave the general structure of the program in pseudo-code:

```
accept a connection from the client
read a single line from the client
if the first word is GET and the second is /
    print "<title>Image gallery from site …</title>"
    print "<h1>Image Gallery from site …</h1>"
for each image file "file"
    print "<img src=$file> <BR>"
else if the first word is GET and the second is the name of an existing image file
    send the image file
else
    print an error page.
Close the connection
```

As usual, your server should be able to accept connections from any number of clients. [Greenberg-3]

The above pseudo-code clearly defines a simple image server. The first few lines say that if the client issues a 'GET /' command to the image server, the server will compose and send back an HTML page of all images; images are specified by the IMG tag and associated URL path of the image. The next few lines say that if the client issues a 'GET <image name>', the server will retrieve and send back the image file that corresponds to that name.

88. Thus in 1996, well before the filing date of the '904 patent, I expected students to be able to build a web-based image server as a matter of course. This was only a part of the assignment, and I do not recollect any student having difficulty doing this.

e. Akamai's FreeFlow Content Delivery System

89. Akamai, Inc. from Cambridge, Mass. business is based around image servers that are separate from the web site that delivers the web page. In particular, they describe FreeFlow as a commercial content delivery system that delivers rich media content, including graphics. As described in their 1999 brochure [Akamai-1], 3rd party companies – Akamai clients – can:

> … migrate all or selected portions of content to be served by the Akamai network with an easy-to-use software utility called FreeFlow Launcher. FreeFlow Launcher tags objects within a Web page that are to be served over the FreeFlow network. When customers request those objects, the Akamai network serves them from the closest-available, highest-performing server rather than from some distant or overloaded server. [Akamai-1, page 1]

The way this works is that the web site owner still serves the original web pages, while objects within the page – rich media such as images, video and animation – are delivered by "a distributed global network of servers that work together to place content close to Web users." Thus FreeFlow's architecture and indeed its business model is based on having image servers separated from the web servers delivering the actual web page. Figure VI-22 below reproduces Figure 3 in [Akamai-1], where it characterizes Internet content delivery with Freeflow. Steps 1 and 2 shows how the user enters the standard URL, and how the customer's web server returns the HtML with embedded URLs pointing to the Akamai Network. The user's browser then requests these embedded objects from the Akamai Network, which returns the rich content by a local Akamai server.

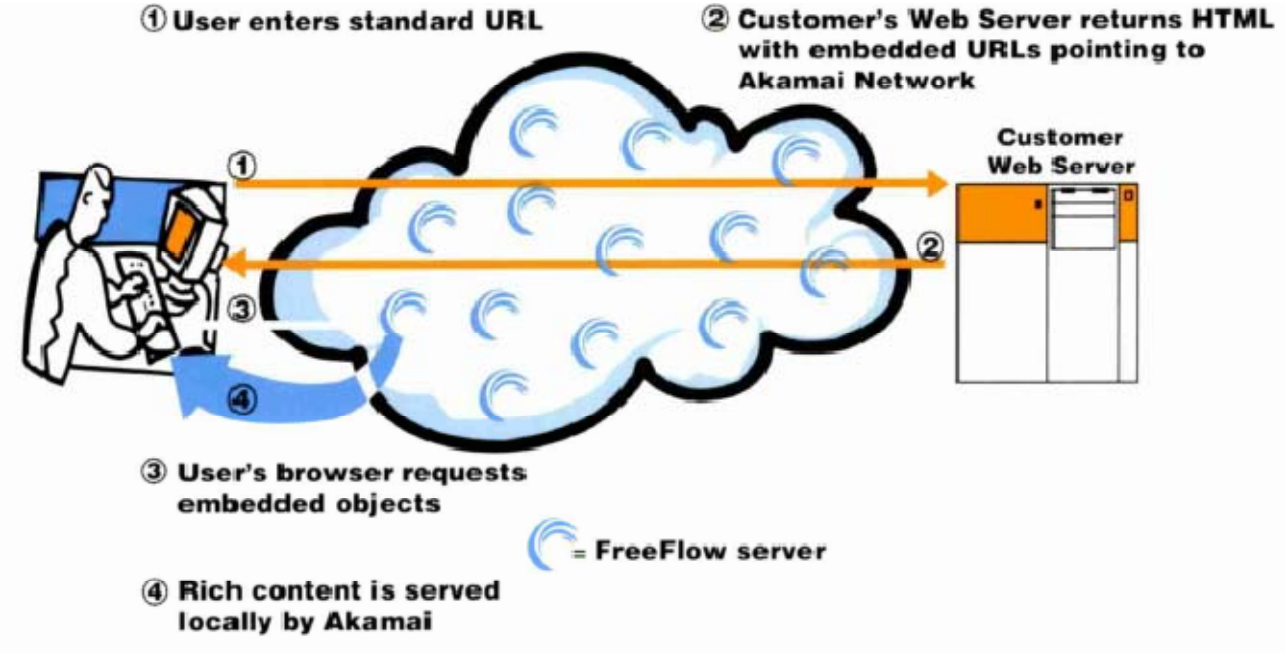


Figure VI-22. Akamai's Freeflow from [Akamai-1]

90. Cisco Systems, Inc. further elaborated on the capabilities of Akamai's architecture, and how it proved effective in delivering multimedia content to a very large audience. As explained in their 1999 white paper, Cisco along with the United Nations Development Program was developing the NetAid web site, where its primary goal was "to raise awareness and foster enhanced communications over the Internet to contribute to the fight against poverty" [Cisco-1, page 1]. Part of this website included three concerts broadcast over the Internet on October 9, 1999. Because this website had to handle large amounts of traffic (more than 40 million hits were recorded during the concert events, with Cisco claiming these being record traffic numbers), Cisco paid special attention to the way information could be effectively and reliably delivered to site visitors. In particular, they used Akamai Technologies servers to manage the site Web graphics: "Akamai Technologies had 1200 servers in place around the world that were available to the NetAid project to accelerate all graphics requests." [Cisco-1, page 3]. Their figure on page 4 explains how Cisco hosted the text portion of the Web

site to ensure speed, quality and efficiency, while Akamai Technologies served the graphics for the Web site content using their distributed network. The effect is summarized on page 7: "the NetAid Web site offloaded graphics requests to Akamai servers to reduce traffic load on the primary Web servers by as much as 90-percent". The Cisco Systems white paper elaborates on their motivation as well as how this worked under the covers [Cisco-1, page 8]:

> The Cisco design team was very concerned about their ability to scale the Web site. Graphics on the site could significantly slow performance. Each Web page at the site had nine to 30 graphics. With thousands of potential visitors downloading the same pages at once, there was a strong potential for delay if all content were stored on a single server. Fortunately, the Web offers a workaround for performance. When a browser first requests a page, it contains a plaintext HTML file with text and page layout and references to inline images. Each image is individually requested by the browser and seamlessly laid out on the screen as a single, integrated "page." These inline references can point anywhere. ¶Inline graphic references on every HTML page in the NetAid Web site were modified during the content-distribution process using the Akamai Launcher script. For example, an HTML tag that looked like:
> <img src=http://www.netaid.org/images/netaidlogo.gif>
> became
> <img src+"http://a292.g.akamaitech.net/7/292/950/2db1947852607d/www.netaid.org/images/netaidlogo.gif>
> A browser calling up a page at the NetAid Web site automatically requested graphics that were served from the Akamai distributed network of 1200 FreeFlow servers around the world....

f. US Patent 6,108,703 (Leighton et. al.)

91. US Patent 6,108,703 title 'Global Hosting System' by Leighton et. al. [US Patent 6,108,703] from the Massachusetts Institute of Technology discloses ideas similar to those implemented by Akamai Technologies. That is, they describe how a content server separate from the web page server can provide HTML page objects to clients such as browsers. As we will see, an image server is just a type of content server. The inclusion of images as part of the content they can handle is detailed in their abstract:

> The inventive framework comprises a set of servers operating in a distributed manner. The actual content to be served is preferably supported on a set of hosting servers (sometimes referred to as ghost servers). This content comprises HTML page objects that, conventionally, are served from a Content Provider site. In accordance with the invention, however, a base HTML document portion of a Web page is served from

> the Content Provider's site while one or more embedded objects for the page are served from the hosting servers, preferably, those hosting servers near the client machine. By serving the base HTML document from the Content Provider's site, the Content Provider maintains control over the content. . [US Patent 6,108,703, Abstract]

The content server they describe covers any embedded content servable as separate objects. This includes, of course images, which Leighton et. al. recognize as part of their content:

> As seen in FIG. 2, a typical Web page comprises a markup language (e.g. HTML) master or base document 28, and many embedded objects (e.g., images, audio, video, or the like) 30. Thus, in a typical page, twenty or more embedded images or objects are quite common. Each of these images is an independent object in the Web, retrieved (or validated for change) separately. The common behavior of a Web client, therefore, is to fetch the base HTML document, and then immediately fetch the embedded objects, which are typically (but not always) located on the same server. According to the present invention, preferably the markup language base document 28 is served from the Web server (i.e., the Content Provider site) whereas a given number (or perhaps all) of the embedded objects are served from other servers. As will be seen, preferably a given embedded object is served from a server (other than the Web server itself) that is close to the client machine, that is not overloaded, and that is most likely to already have a current version of the required file. [US Patent 6,108,703, col. 5:23]

Leighton et. al. offer various advantages to content servers, which would also apply to image servers. A few are listed below.

> It is a general object of the present invention to provide a computer network comprising a large number of widely deployed Internet servers that form an organic, massively fault-tolerant infrastructure designed to serve Web content efficiently, effectively, and reliably to end users.
>
> Another more general object of the present invention is to provide a fundamentally new and better method to distribute Web-based content. The inventive architecture provides a method for intelligently routing and replicating content over a large network of distributed servers, preferably with no centralized control.
>
> Another object of the present invention is to provide a network architecture that moves content close to the user. The inventive architecture allows Web sites to develop large audiences without worrying about building a massive infrastructure to handle the associated traffic.
>
> Still another object of the present invention is to provide a fault-tolerant network for distributing Web content. The network architecture is used to speed-up the delivery of richer Web pages, and it allows Content Providers with large audiences to serve them reliably and economically, preferably from servers located close to end users.

> A further feature of the present invention is the ability to distribute and manage content over a large network without disrupting the Content Provider's direct relationship with the end user.
>
> Yet another feature of the present invention is to provide a distributed scalable infrastructure for the Internet that shifts the burden of Web content distribution from the Content Provider to a network of preferably hundreds of hosting servers deployed, for example, on a global basis.
>
> In general, the present invention is a network architecture that supports hosting on a truly global scale. The inventive framework allows a Content Provider to replicate its most popular content at an unlimited number of points throughout the world. As an additional feature, the actual content that is replicated at any one geographic location is specifically tailored to viewers in that location. Moreover, content is automatically sent to the location where it is requested, without any effort or overhead on the part of a Content Provider. [US Patent 6,108,703, col. 2:40]

g. Ayers and Stasko's MosaicG

92. As part of the previously described MosaicG system, Ayers and Stasko describe its implementation. For pragmatic reasons which they explain in Section 4 of [Ayers-1], they modified the source of the Mosaic 2.5 browser to accommodate MosaicG. However, their description anticipates a separate image server. It stores thumbnail visual images as well as other information that comprise part of the MosaicG tree structure and specifically say that this storage is separate from the NCSA web browser:

> History information is stored in a hash table and a tree structure that is separate from NCSA Mosaic's internal data structures. … [Ayers-1, Section 4]

h. Card, et. al.'s Web Book / Web Forager

93. As part of the previously described WebBook/Web Forager system, Card et. al. describe the importance of caching and storing images locally, i.e., they describe an image server separate from the web site that contains the original pages. While they do not provide details, they state its motivation [Card-1, page 115]:

> … The HTML image on the new page develops at the slow Internet speeds (often 15~30 sec). Web pages entered into the space are stored locally and thence forward

are available at user interface speeds (around 1 ~0.1 sec), permitting high interaction rates. These pages can also be grabbed and placed into WebBooks.

i. Wittenburg et. al.'s Polynav

94. As part of the previously described Polynav system, Wittenburg et. al. teaches a thumbnail image server that collects, renders, stores and serves images culled from web page URLs. They lay the foundations for such an image server in [Wittenburg-1, page 78], where named URLs are associated with images:

> Preparation for previewing an information space through one of the PolyNav clients involves a process of acquiring, structuring, and storing meta-information for the objects of navigation. So far we have concentrated on acquiring images from HTML Web pages; however, we have also extracted associations of images with URLs from databases to handle dynamically generated information and have experimented with other kinds of metadata.

They then state why such an image server makes sense within this context.

> While it may be that in the future all of the necessary image metadata will be available in real time on the Web, that is not the case today. Automatically acquiring the metadata necessitates parsing HTML for each URL (unless the URL/ image association is available through a database query), filtering the images contained therein, and then downloading and preprocessing the images to address computation and bandwidth constraints.

95. Polynav's architecture specifies and implements a thumbnailer and an image server, illustrated in Figure VI-23 below (taken from Figure 3 in [Wittenburg-1]).Wittenburg et.al. explains how this works [Wittenburg-1, emphasis added] . The quote below illustrates some of this, although additional details are found in their paper.

> …users can specify Web spaces through queries or Web walks. Our query service makes use of two external search engines: one a popular search engine available on the Web; the second an experimental metasearch service at GTE Laboratories. The PN_Server passes the query on to a search engine, parses the query results, and creates a PNF file. The PNF file is then processed by a service that creates individual image thumbnails as well as collage thumbnails (all images from each page URL included in one image). The PNF file is now ready for use by one of the two PolyNav preview clients, or it may be processed further for the purposes

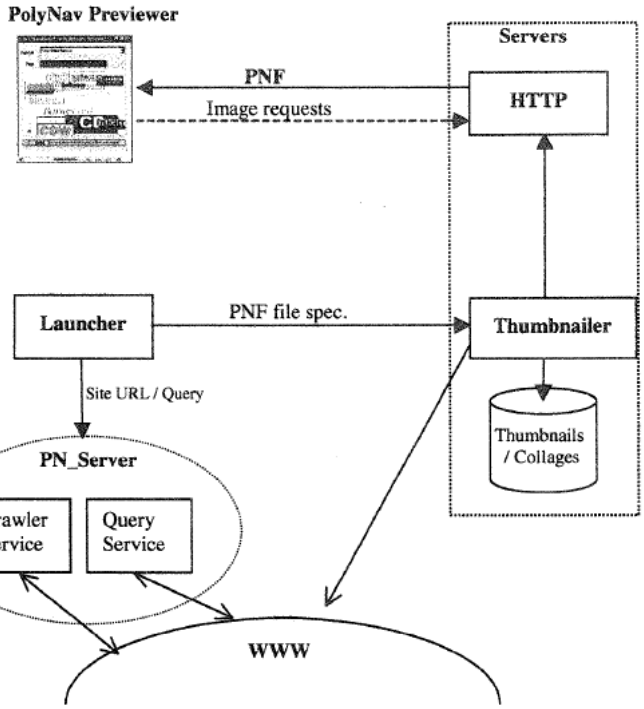


Figure VI-23. Architecture of PolyNav, from [Wittenburg-1]

> of individual applications. The Web crawler service follows links originating from one or more start URLs and generates a PNF file to a specifiable depth.
> …
> In the current prototype, **we pregenerate thumbnail images that are used in place of the images and store these thumbnails within our server**. **Thus we eliminate the need to download an image from remote servers more than once** and reduce the size of the data that needs to be transferred to the navigator.

Finally, Wittenburg et. al. also describe how they can improve efficiency of this image server through a second image server that behaves as a cache [Wittenburg-1, emphasis added]:

> We have begun to explore how to better support this kind of application by using a cache together with the thumbnail generation service. When an image is requested by a navigator via the corresponding thumbnail URL**, the server that gets the request is really a caching server that acts as a front for the real thumbnail server. If the requested thumbnail is still in the cache from a previous request, it is returned immediately.** Otherwise, the caching server forwards the request on to the real thumbnail server which generates the thumbnail at that time just as described above. The initial generation of each thumbnail does take significant time, but they are then cached for future access. Thumbnails that are not used as much may be flushed from the cache to make room for others.

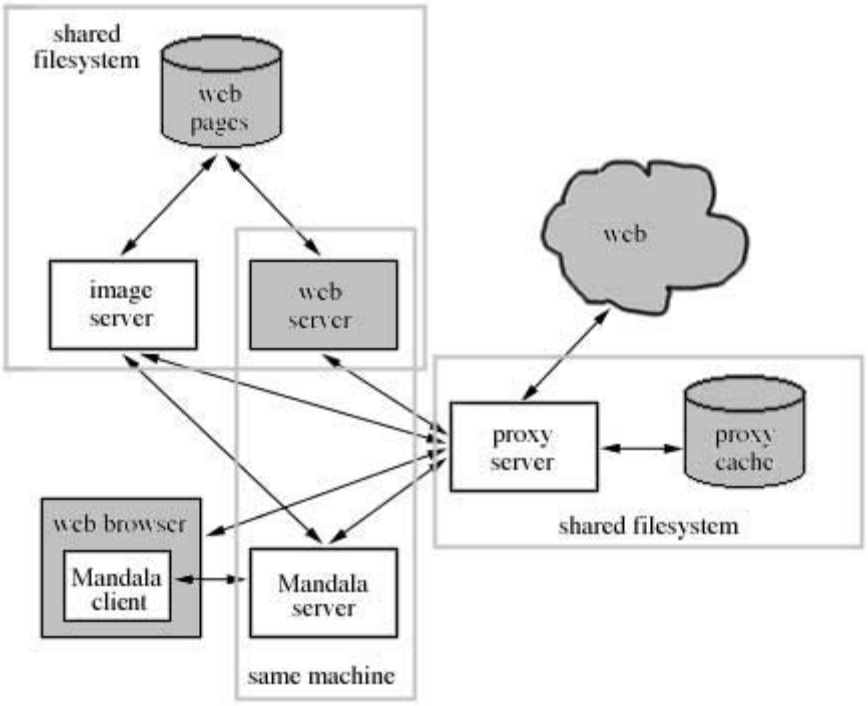


Figure VI-24. Mandala's Component Structure, from [Helfman-1]

j. Mandala, the Imago Image Server, and Mirage, by Helfman

96. I previously described how in 1999, Jonathan Helfman's Mandala embodied yet another system that used thumbnail images to represent, access and organize web information [Helfman-1]. His paper also describes Mandala as an architecture which, as we will see, includes a separate thumbnail image server called Imago. To repeat part of our previously presented quote: "Mandala's image server, called Imago, has been developed to support fast image data compression and decompression, image shrinking, imagemap creation, and image meta-data extraction … Imago creates thumbnails and imagemaps according to client specifications." [Helfman-1, pages 165-66]. Helfman details his architecture and implementation in Section 2; which includes Figure VI-24 above as reproduced from Figure 2 in [Helfman-1]. Note that the text and the arrows in the figure illustrates that the separate components, including the image server in the upper right corner, communicate to each other via an HTTP-like message protocol.

> 2 Architecture and Implementation. Figure 2 illustrates Mandala's component structure. White boxes represent the Mandala components. Solid grey shapes represent standard components that are used without modification (e.g. web servers and browsers). Light grey boundaries indicate runtime constraints. Arrows indicate the exchange of HTTP-like messages consisting of ASCII headers and optional ASCII or binary data. [Helfman-1, page 165]

97. Helfman's architecture also contains a 2nd image server called Mirage, which is a separate proxy server that creates and serves a local cache of images (right side of figure):

> "Mandala's proxy server, called Mirage, has been developed to support local caching of images, transparent web browser monitoring, and HTML parsing. A proxy server is a program that sits between web servers and web clients, such as browsers . . . . Mirage improves Mandala's performance by caching thumbnails and imagemaps generated by Imago and installed on a web server." [Helfman-1, pages 166-67]

k. Sclaroff et. al.'s ImageRover

98. In 1999, Sclarrof, Taycher and LaCascia from Boston University described and built a system for generating an image server database of thumbnail images gathered from web pages [Sclaroff-1]. Sclaroff et. al. write:

> As shown in the Figure 1, robots can contain collection modules, digestion modules, and a local database. The collection modules recursively parse and traverse WWW documents, gathering images. The digestion modules then process these images to extract needed image indexing information and to compute a reduced resolution thumbnail image. The robots are dispatched and coordinated via a separate coordination layer, which also manages updates of the image index database. [Sclaroff-1, Page 3]

99. Essentially, Sclarfoff et. al.'s architecture comprises robots (or web crawlers) that act as a multiplicity of downloaders that gather images, render these images as thumbnails, and stores them in a database so they can be served by an image server. They explain how this multiplicity of downloaders could be done via threads within a robot, by invoking a fleet of robots running on the same machine, and a fleet of robots running on different machines. Sclaroff et. al. also describe how its collection module recursively parses and traverses

WWW document, which generates the list of URLs that should be used to retrieve images from those URLs.

100. Sclaroff et. al. also justify why a multiplicity of downloaders are useful, where he talks about the time it takes a single robot to collect images vs. the time it takes a fleet of robots to collect images:

> "On average, each single-threaded robot can collect 1044 images daily. It is therefore reasonable to expect that a modest fleet of 32 single-threaded robots can collect approximately 1 million images monthly. Multi-threaded robots should achieve significantly greater throughput." [Sclaroff-1, page 7]

l. Kopetzky et. al.'s Visual Link Previews

101. I already described how, in 1999, Kopetzky and Muhlhauser [Kopetzky-1] described and built a system that creates visual previews of links, including thumbnail visual images of web paes, on a web page for view by end users. Kopetzky et. al. recognized that it would be unreasonable for authors to change their pages to include thumbnail previews. They then describe an architecture, including a separate thumbnail image server, that collects, renders, and stores thumbnail visual images so they can be delivered to the client on demand [Kopetzky-1, pages 1528-30]. They implemented their thumbnail image server as a proxy server that is separate from the browser and from the web site serving the original web page. They describe it as follows, as well of the advantages of such an image server.

> To solve this problem an approach using a proxy server was chosen. An overview of the main components of the proxy server, which has been implemented in Java, can be seen in Figure 7.
> ¶The proxy server has the following tasks:
> *analyze the links in the requested HTML document and generate the preview images for all links in the document;
> *cache the requested HTML documents and the computed link preview images for future access;
> *modify the HTML documents in a way that the requesting browser is able to show the link preview images.

> ¶This approach has the following advantages:
> *the proxy server has to generate the preview information only once (depending on server space);
> *many readers can share one proxy server and thus benefit from already generated preview information;
> *readers only have to make one change in their browsing environment: they have to configure the Web client to use a proxy server – everything else is done automatically;
> *the proxy server can use other proxies servers and thus benefit from information already fetched from the Web. [Kopetzky-1, pages 1528-29]

The text following the above description, although not repeated here, clarifies that the proxy server is separate from both the browser asking for a requested document and from other servers that deliver the source document, i.e., as with most proxy servers, it acts as a separate intermediary between the two.

102. Kopetzky et. al. also disclose that other configurations of image servers are possible. In particular, they describe how a 'Link Service' that provides link information to clients separate from services that deliver the page contents can be extended to include thumbnails:

> 6.4 Link services. An interesting approach to enhance usability of the World Wide Web is the use of a link service [7]. Via link services readers can gain a surplus of information by applying different sets of links to one and the same Web document. Link services store link information external to the documents linked. Current implementations (as shown in [8]) do not store thumbnail information along with link information. We think that by storing the preview image link services could gain attractiveness. [Kopetzky-1, page 1531]

m. Schmid et. al.'s CoBrow

103. Schmid et. al. teaches *a Thumbnail Service* based on a client-server model, which behaves as an image server that delivers thumbnail visual images. In their words [Schmid-1, page 2]:

> A client requests thumbnails of arbitrary Web pages. After the request is received, the server first checks its memory and disk caches. If the requested Web page thumbnail is already rendered and not expired, it is immediately sent back to the client. Otherwise, the server loads the requested Web page and renders the image. There are several possible ways to achieve the render process. In our implementation, we use an existing Web browser, e.g. Netscape, to render the page. Then, the server captures the rendered image and scales the bitmap according to width and height of the client request. Afterwards, the image is encoded according to JPEG [5]. The new generated thumbnail is stored in the server's caches and sent in reply to the client. In order to provide thumbnails concurrently to multiple clients, the server must be capable to process thumbnails simultaneously.

Schmid et. al. also describe the protocol between the client applications and Thumbnail server, where its syntax is given as: request ::= http: //<server name>{:<port>]/?url=<url>{&options?} . That is, this HTTP request from the client includes the server's name and port number, the url of the web page thumbnail that is being requested, and options such as the desired thumbnail size.

104. Figure VI-25 below reproduces Figure 2 from [Schmid-1], which is labeled by Schmid et. al. as the Thumbnail Server Architecture. As seen in the figure, various clients, such a web browser or a custom VRML client, can request a thumbnail service. The request is sent to the Thumbnail Server Host, which either gets a copy of the thumbnail from its cache store, or it renders the thumbnail (by grabbing and scaling the image of the web page captured by the Netscape Browser) if it is not in the cache.

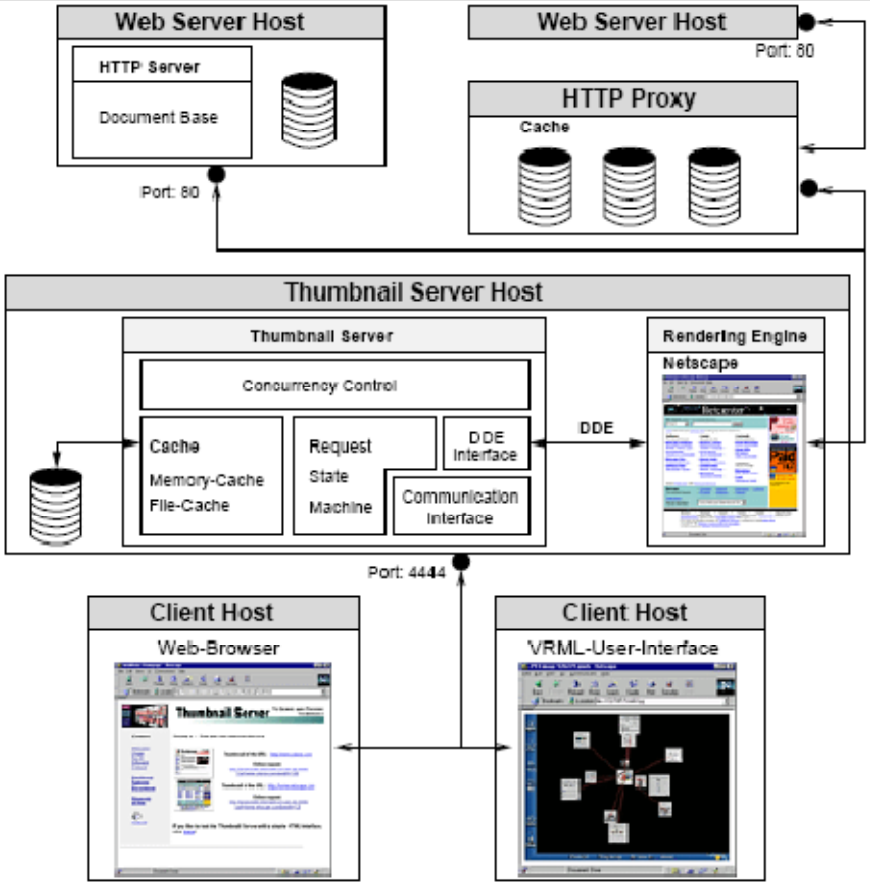


Figure VI-25. Thumbnail Server Architecture, from [Schmid-1]

n. US Patent 6,356,908 (Brown et. al.)

105. I previously described how Brown et. al.'s patent titled *Automatic web page thumbnail generation* anticipated presenting web page content as thumbnail visual images. As well, his patent describes several variations of an image server that stores and provides these thumbnail visual images.

106. First, one of the three methods he describes to produce iconic representations of web page links includes a separate image server. He says: "Finally, the third alternative method for producing these icons is to send the information via a separate protocol/communication with the server" [US Patent 6,356,908, col. 8:35]. Although a brief description, one of ordinary skill in the art would have understood that this sending of

information via a separate protocol/communication describes a client/server relationship, i.e., a client and separate image server.

107. Second, Brown discloses employing a proxy server as an image server. The in particular, he describes a "Thumbnail Assistant" that:

> "intercepts and parses documents after communications 510 receives documents, but prior to processing by language interpretation 512. After parsing documents, thumbnail assistant 516 generates thumbnail images of linked pages to a loaded web page and displays these thumbnails to a user on a client machine. . . . Furthermore, **thumbnail assistant 516 could be utilized on a proxy server, wherein thumbnail assistant 516 pre-generates web pages on a web server prior to receipt by browser 500.** In this instance, which would be preferred for legacy browsers, the server would modify the stream sent to the user and place references to the thumbnails in that stream." (emphasis added) [US Patent 6,356,908, col. 5:17-52, emphasis added]

Steps 725, 730, 735 and 740 in US Patent 6,356,908 Fig. 7 illustrates how this process works in the preferred embodiment, where thumbnails are stored in the cache (the image server) as they are generated:

> "If the thumbnail option has been selected, then thumbnail assistant 516 parses the web page for links to other web pages (step 725). Thumbnail assistant 516 then checks the cache for linked pages and prefetches the linked pages that are not in the cache (step 730) using the prefetch mechanism associated with web browser 516. **Thumbnail assistant 516 then generates thumbnails of each linked page that does not already have a thumbnail in the cache (step 735) and then stores the newly generated thumbnails in the cache** (step 740)." (emphasis added) [US Patent 6,356,908, col. 6:17-26]

108. Brown's description of the thumbnail assistant also defines an image server that is logically separated from the web server. In particular, in US Patent 6,356,908 Figure 5, he illustrates an esmbodiment that has a separate "Communications (HTTP)" block that receives documents, a separate "Thumbnail Assistant" that further processes the document to generate and display thumbnails, and a separate "Graphical Display" that presents web pages to the user as described below. One of ordinary skill in the art would have understood that the thumbnail

assistant that contains the image server was a logically distinct entity. One of ordinary skill would also understand that the Communications block, part of the browser, was communicating via HTTP, a standard internet protocol, with a web server, while getting those pre-stored thumbnail visual images from the cache.

109. While Brown discloses using the Thumbnail Assistant as a browser plug-in, (col. 5:37-45), he also discloses using the Thumbnail Assistant on "a proxy server, wherein thumbnail assistant 516 pre-generates web pages on a web server prior to receipt by browser 500." [US Patent 6,356,908, col. 5:46-49]. Thus Brown discloses yet another type of image server in this embodiment, i.e. the Thumbnail Assistant residing on its own server and providing images to the separate web browser, and which is separate from the web server that supplies the underlying web page.

o. US Patent 7,177,948 (Kraft et. al.)

110. I previously described how Kraft et. al.'s patent titled *Method and apparatus for enhancing online searching* [US Patent 7,177,948] anticipated presenting web page content as different-sized thumbnail visual images. His patent also describes an image server that stores and provides these thumbnail visual images.

111. First, Kraft discloses using an server that generates and provides the medium-sized thumbnail visual image, where the thumbnail is stored in a cache database:

> "**Generation of the medium sized thumbnail (also called the medium sized visual abstract) is preferably done on the server side. The server preferably uses a caching mechanism to store the medium sized visual abstracts in a cache database so that users who later access the same document need not regenerate the medium sized abstract.** The database may be programmed to store the medium sized visual abstract for a specific amount of time and then delete the abstract to conserve space." [US Patent 7,177,948, col. 4:50-58, emphasis added]

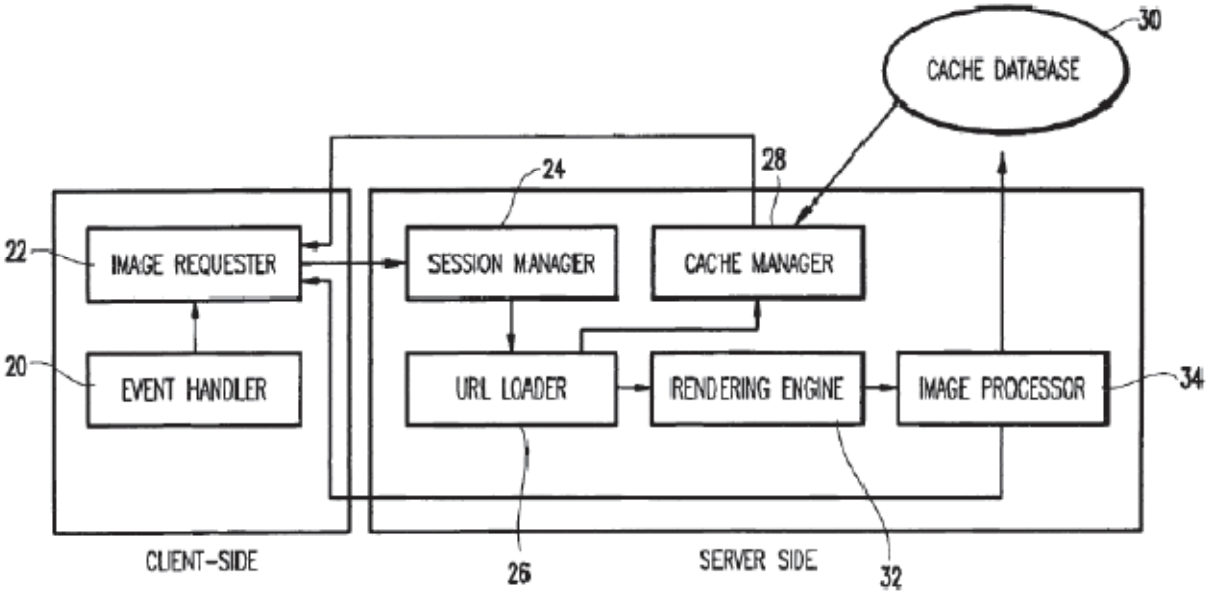


Figure VI-26. Figure 4 from [US Patent 7,177,948]

US Patent 7,177,948 in Fig. 4, reproduced here in Figure VI-26 above, clearly labels and illustrates how this server side thumbnail provider (including the cache) is separate from the client-side image requestor. The text associated with that figure explains how this works (but see also col. 6:1-41 for further details)

> "The system is preferably implemented as a distributed client-server application as described below with respect to FIG. 4. This disclosed system is not limiting as other systems that perform the above disclosed method are also within the scope of the present invention. . . . On the client side of the system, the event handler 20 tracks actions of the user. . . . If a user moves the mouse pointer over a specific spot on a result item, or preferably over a (small) visual abstract 14, the event handler 20 triggers an event to the image requester 22 that contains the result item number/id (e.g., document number) and the URL of the requested documents. . . . **The image requester 22 requests the medium sized thumbnail 16 of a document from the server.** . . . The above-described client-side components and their basic functionalities are already integrated into most modern web browser technologies. These web browsers provide an application programming interface (API) for scripting languages to achieve the functionalities discussed above. . . . **The server-side components interact closely to achieve the desired result**. . . . **The URL loader 26 looks to the local cache (i.e., cache database 30) by asking the cache manager 28 whether a medium sized thumbnail 16 for the requested document is already stored in the cache database 30.** This saves time and increases the overall performance of the system. The system may also include additional component(s) that

> detect idle cycles of the system and then uses these to generate the medium sized thumbnail 16 in advance." [US Patent 7,177,948, col. 5:39-6:21, emphasis added]

Kraft further motivates explains their thumbnail visual image cache manager and cache data base (i.e., an image server):

> "For performance reasons, the URL loader 26 asks the cache manager 28 whether the desired URL was previously loaded. In this case it can directly retrieve the rendered and captured image from the cache manager 28 and pass the visual abstract to the representation manager 62. This saves a lot of work and time and therefore speeds up response time. . . . [T]he cache manager 28 stores image thumbnails (i.e., visual abstracts) in a cache database 30 and keeps tracks of the rendered documents along with a time stamp for each resource. Before the time intensive process of rendering and image processing is initiated, the system first queries the cache manager 28 to determine whether the document is already processed. If so, then the cache manager 28 simply returns the visual abstract." [US Patent 7,177,948, col. 8:17-51]

p. US Patent 6,058,417 (Hess)

112. I previously described how Hess et. al.'s patent titled *Information Presentation and Management in an Online Trading Environment* [US Patent 6,058,417] anticipates the use of thumbnails as well-known general technique to enhance the meaning and recognizability of a link, rather than a method restricted only to web pages. As a generalizable technique, Hess also teaches image servers within this context. Figure VI-27 below reproduces Figure 4 from US Patent 6,058,417, described by Hess as a high level illustration of the interaction among various devices comprising their online commerce site. Their illustrated architecture describes a thumbnail image server, which includes a thumb building machine, a separate thumb server that clients interact with to receive image data, and a separate thumb database holding all the thumbnail images:

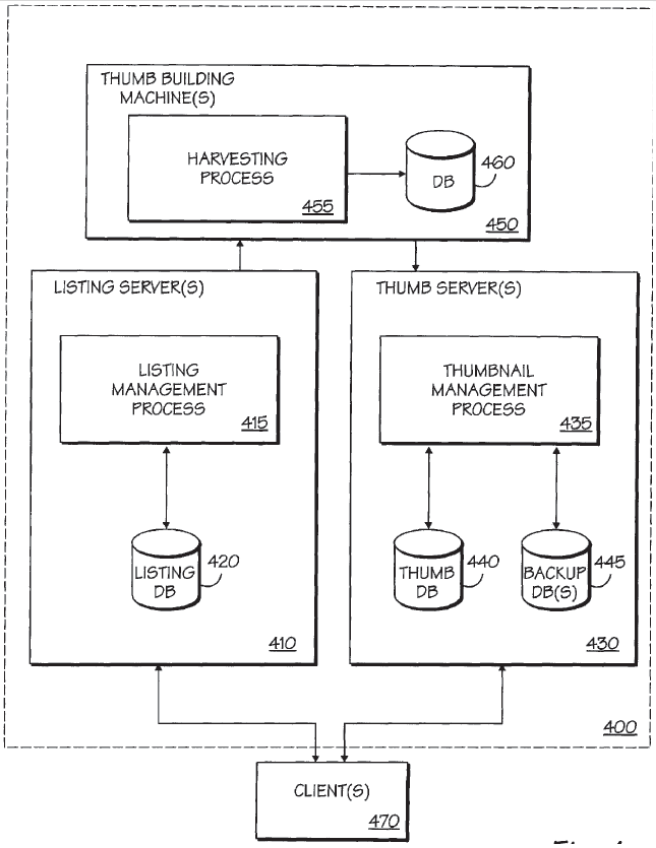


Figure VI-27. Figure 4 from [US Patent 6,058,417]

> The thumb building machine 450 includes a harvesting process 455 and a database 460. As will be described further below, the harvesting process 455 periodically harvests images that sellers have associated with items in the listing database 420. After a set of images have been harvested and thumbnailed, the harvesting process 455 notifies the thumb server 430 that new thumbnails are available.
>
> The thumb server 430 includes a thumbnail management process 435, a thumb database 440 and one or more backup databases 445. Clients 470 interact with the thumbnail management process 435 to receive image data associated with the Gallery format. When new thumbnails are available, the thumbnail management process 435 makes a backup copy of the current thumb database 440, receives a copy of a new database from the thumb building machine 450, and begins serving thumbnail images from the new database.
>
> Importantly, as one feature of the present embodiment, thumbnail images are not stored as individual files; rather, they are stored in an efficient database format that will be described further below. [US Patent 6,058,417, col. 5:25-45]

Hess discloses further details about how this system works. This includes the harvesting of images from a plurality of sites based on user-supplied information, that image location

information of the set of images to be downloaded is a URL, that multiple images downloads are started concurrently, and that the thumbnail database is designed for rapidly retrieving information. Hess specifically discloses that the various servers shown in Figure VI-27 are separate:

> Note that in this description, in order to facilitate explanation, the thumb building machine 450, the listing server 410, and the thumb server 430 are generally discussed as if they were each a single **device. However, each of the thumb building machine 450, the listing server 410, and the thumb server 430 may actually comprise multiple physical and/or logical devices connected in a distributed architecture, and the various functions performed may actually be distributed among multiple devices.** Additionally, in alternative embodiments, the functions performed by the various servers may be consolidated and/or distributed differently than as described. For example, any function can be implemented on any number of machines or on a single machine. Also, any process may be divided across multiple machines. [US Patent 6,058,417, col. 5:61, emphasis added]

q. Ad Servers as Separate Image Servers: Langheinrich et. al., Double Click, US Patent 5,948,061 (Merriman)

113. Before the ‘904 patent filing date, separate image servers were also well-established in other Internet-related areas, including web advertisement. In web advertisement systems, a client process asks an image server (an ad server) for an advertisement such a banner ad. The server returns the advertisement image and the client inserts it into the web page as a clickable image. The various examples below, culled from the academic, patent and corporate literature and produced in the era before the ‘904 patent, illustrate this basic process.

114. Langheinrich et. al. [Langheinrich-1] described and illustrated the state-of-the-art of ad servers in his May 1999 paper *Unintrusive customization techniques for web advertising.* Figure VI-28 below reproduces Figure 2 in [Langheinrich-1], and the text and captions below describe how this works. Note that he describes an advertisement as a banner images, verifying that an ad server is a type of image server.

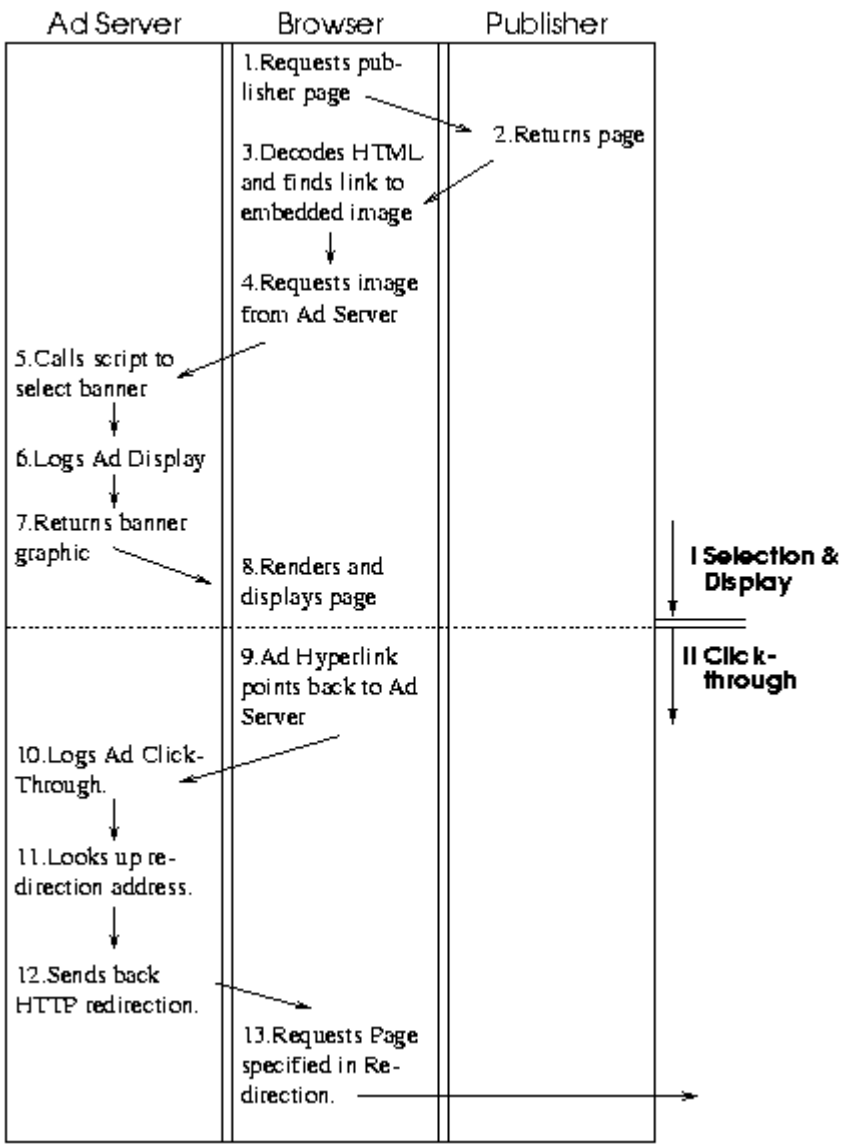


Figure VI-28. Figure 2 from [Langheinrich-1]

> The basic process is outlined in figure 2: the Web page of an online service (the "publisher") contains a link to a banner advertisement. Although the content of the original Web page (step 2) stays the same, the Ad Server will potentially select different banner images for subsequent advertisement requests (steps 5-7). ¶Once a user shows interest in the displayed banner and clicks on it, the surrounding hyperlink will point back to the Ad Server where a script notes the click and sends a redirection message with the correct advertiser's site back to the originating browser. [Langheinrich-1, Section 2.1]

The caption for Figure 2 (shown in Figure VI-28) stresses that the ad server is separate from the web page server, and that the advertisement image contains a link to a site:

> Figure 2: Online Advertisement - Control & Data Flow**. Embedded HTML images (step 3) and HTTP redirection (step 12) make it possible to separate page content and advertisement.** After obtaining the page content from the publisher's Web server, the user's browser will request a dynamically selected advertisement from the Ad Server (part I). O**nce the user clicks on the advertisement (part II), the Ad Server will redirect the user to the appropriate site**. [Langheinrich-1, Figure 2 caption; emphasis added]

115. Merriman et. al.'s patent titled *Information Method of delivery, targeting, and measuring advertising over networks* [US Patent 5,948,061], filed on October 29, 1996, also describes a separate ad server as image server.

> To effect such a capability, **an advertising server process is provided as a node on the network. The various advertisements are stored on the network of the server and preferably on the server.** When, for example, a user using a web browser accesses a web page that is affiliated with the advertising server process, the affiliated page's encoding includes an embedded reference to an object provided by the advertising server process. **That causes the user's browser to contact the advertising server process to provide the advertising image or information that will appear on the accessed web page as displayed by the user's browser.** [US Patent 5,948,061, col. 2:15, emphasis added]

US Patent 5,948,061 stresses the separation of the ad server, that the ad server returns images, and that the images contain a link. Indeed, the process it describes is very similar to the state-of-the-art as described by Langheinrich et .al. in 1998 above.

> The basic operation of the system is as follows in the preferred embodiment. When a user browsing on the Internet accesses an affiliate's web site 12, the user's browser generates an HTTP message 20 to get the information for the desired web page. The affiliate's web site in response to the message 20 transmits one or more messages back 22 containing the information to be displayed by the user's browser. In addition, an advertising server process 19 will provide additional information comprising one or more objects such as banner advertisements to be displayed with the information provided from the affiliate web site. Normally, the computers supporting the browser, the affiliate web site and the advertising server process will be at entirely different nodes on the Internet. Upon clicking through or otherwise selecting the advertisement object, which may be an image such as an advertisement banner, an icon, or a video or an audio clip, the browser ends up being connected to the advertiser's server or web site 18 for that advertisement object. [US Patent 5,948,061, col. 3:5, emphasis added]

This separation is repeated elsewhere in US Patent 5,948,061. Figure 1 in the patent (not reproduced here) shows a separate advertising server process, a separate advertiser's web site, a separate affiliate web site, and a separate user's browser, all who interact via the HTTP protocol. The text describing Figure 1 emphasizes this separation, and also states that this process uses the IMG tag standard as presented earlier to specify and retrieve these advertising images:

> These objects preferably do not reside on the affiliate's web server. Instead, the affiliate's web server sends back a link including an IP address for a node running an advertiser server process 19 as well as information about the page on which the advertisement will be displayed. The link by way of example may be a hypertext markup language (HTML) <img> tag, referring to, for example, an inline image such as a banner. The user's browser 16 then transmits a message 23 using the received IP address to access such an object indicated by the HTML tag from the advertisement server 19. [US Patent 5,948,061, col. 3:34]

Merriman also emphasizes that images are stored in a database:

> Each advertisement along with a table of the targeting profile criteria for the advertisement and other data regarding the advertisement currently available is stored in a database such as shown in FIG. 3B. The actual advertising object, which may be a banner image in a GIF or JPEG file format, an icon for an audio or video clip or some other object is kept as part of the advertising server process. [US Patent 5,948,061, col. 5:64]

116. DoubleClick was a company that, in 1996, commercialized an image-based ad server through its DART 'Dyanmic Advertising Reporting & Targeting' ad serving and reporting service [Doubleclick-1]. The various documents available on the DobleClick web site in early 1998 described its basic capabilities [DoubleClick-1 to 4]. DoubleClick-4 explains:

> How does DART Do It?
> When a user visits a DART-enabled Web site, the selected Web page is delivered from that Web site to the User, and the User's browser loads the requested page. Embedded in the page are Image Tags that link the browser to the closest DoubleClick DART ad server. The User's browser initiates the HTTP Get request from the server, establishing a connection between the two, and a graphic file is requested from DART to fill the ad banner space on the Web page being loaded on the User's screen. [DoubleClick-4]

This explanation clearly describes an image server that is separate from the user's browser and from the web server delivering the web page. One of DoubleClick's web pages [DoubleClick-3] then details in pseudo-code how a web site would use DART. Included in this explanation is that it includes: embedding commands to the web browser to download, via the image server, images, and using the standard IMG tag in a manner consistent with how I had previously explained it. As well, they state how the returned image also contains a link:

> "When the user clicks-through, the ad servers receive a Get request and redirect the User's browser to the URL of the site that placed the ad banner. The advertiser's server delivers the page content…" [DoubleClick-4]

DoubleClick also reports on various advantages of using such a server arrangement, several which are quoted below [DoubleClick-1].

1. "Ad Delivery. DoubleClick helps sites avoid the costly investment of purchasing and maintaining ad servers by using a centralized ad serving solution. Sites simply include HTML tags on their pages and DoubleClick matches an appropriate targeted banner, in less than 20 milliseconds."
2. "This centralized solution allows sites to take advantage of our ability to aggregate and target based on user data. Soon, this will allow us to deliver demographically targeted ads."
3. "DoubleClick's systems are extremely redundant and have operated at 99.9% uptime since 1/97" [DoubleClick-1]

r. Summary

117. The above background is an illustrative rather than exhaustive overview of how image servers have been developed and used by many others in ways that anticipate various purportedly inventive elements of the '904 patent. We have seen that:

- The world wide web incorporates a special IMG tag that enables any web server to behave as a separate image server as a matter of course;

- The world wide web incorporates a standard – the Common Gateway Standard – that allows any client to communicate with specialized servers, including those that deliver content stored in databases;
- Constructing a web-based image server was well within the expected skills of undergraduate computer scientists (e.g., as shown in my course assignment in 1996);
- Many inventors knew that rendering, storing, and retrieving thumbnail visual images was a basic part of their system (e.g., [Ayers-1; Card-1; Wittenburg-1]);
- A variety of different architectural styles for image servers had been developed, e.g., as part of a cache manager, as part of a proxy server, as part of a web page server, etc.;
- Separate image servers specifically designed to efficiently serve images separated from other web page content were patented (e.g., [US Patent 6,108,703; US Patent 5,948,061]), commercially available (e.g., [Akamai-1; Cisco-1; DoubleClick-1 to 4]), and justified in accompanying literature (e.g., [Langheinrich-1; US Patent 5,948,061; Akamai-1]).
- Image servers were applied to store a variety of image types, ranging from any image (e.g., [Andreesen-1; US Patent 6,108,703; Akamia-1; Cisco-1]), to thumbnails of any image (e.g., [US Patent 6,058,417]), to thumbnails of images contained within web pages (e.g., [Wittenburg-1; Helfman-1], to thumbnails of the web page itself (e.g., [Schmid-1; Kopetzky-1; US Patent 6,356,908; US Patent 7,177,948].

C. **SPLITTING, TRIMMING, AND CONSTRUCTING URLS**

118. The notion of splitting, trimming and constructing URLs, or indeed any path (such as a file path) is an elementary skill that any undergraduate student in computer science circa the time of the patent would know how to do this (and likely has done this as part of

undergraduate programming assignments). This topic generally falls under the heading of 'parsing' and/or 'string manipulation', which is a standard topic in computer science. Thus I will not describe its history at any length, except to show a few examples that splitting and trimming was a common function of many programming languages, and that trimming of URL paths was well known in the art in 1999 and earlier.

a. Greenberg's Assignment 3

119. I previously mentioned how, in 1996, I gave my CPSC 441 Computer Communications class comprising 3rd and 4th year undergraduates an assignment titled 'Building An Image Web Crawler and a Web Server.' The course assignment as given out is in [Greenberg-3]. The assignment contained 3 parts, where the assignment specifications says: "The more parts completed, the more marks you get". The first part directs students to build 'a multhreaded image grabber client for HTML pages. The specifications specifically state:

1. Program use: image_grabber URL:PORT
e.g., image_grabber /www.cpsc.ucalgary.ca/home.html:80
note: to save parsing, you can also make this 3 separate arguments
2. Result: all images in that URL are saved as individual files in your directory
3. Client behavior

- parse the server name and directory name from the argument
- connect to the server
- if a connection to the server cannot be made, print an error message an abort
- as the page is being read via sockets, look for HTML img directives and the source of the file, eg <img src="/images/gtr.gif">
- create a new path for the src if it does not have a leading '/' [Greenberg-3]

Notice that the assignment directs students to parse, or 'split' and thus trim the server name from its directory path, and to 'construct' a new URL for a relative image path (i.e., on that does not have a leading '/'). Thus in 1996, well before the filing date of the '904 patent, I expected

students to be able to split, trim and construct URLs as a matter of course. This was a very minor part of the assignment, and I do not recollect any student having difficulty doing this.

b. PHP and Java support for splitting, trimming and constructing strings

120. Almost every programming language includes functions to manipulate strings so that they can be split, trimmed and reconstructed from various parts. For brevity, I will illustrate this using the PHP programming language version 2.0 (circa 1996) which was developed specifically for the Web and manipulating web page contents. As defined in Version 2 by the PHP Group [PHP-1], PHP has a variety of built-in functions for manipulating strings and links. For example, trimming and splitting is easily done through its 'substr', 'strchr', and 'strstr' functions that return part of a given string, as well as PHP's various regular expression functions that provide for more complex string manipulation. The 'strtok' will tokenize a string, that is, split it into individual strings as defined by a particular separator or token (e.g., the '/' character in a URL). The 'sprintf' function will join various strings together to form a new one. Later versions of PHP (still within the time frame of the patent) had even more functions for convenient string manipulation of this type, e.g., by manipulating string elements as an array. The URLs that specify links are, of course, just a type of string whose regular format makes it particularly easy and predictable to split, trim and construct (e.g., from its the use of '/' and '//' as delimiters).

121. The Java language, designed with the Internet in mind, has a URL class that simplifies parsing of URLs into its components even further. Campione and Walrath's 1996 book "the Java Tutorial" is one of the many practitioner's texts at the time that described this class [Campione-1]. They describe how a URL comprises a protocol (the protocol identifier),

host name (the name of the machine the resource lives on), a file name (the pathname to the file on the machine), the port number (the optionally provided port number to connect to), and the reference (a named anchor within resource or file) [Campione-1, page 45]. Their section titled "Parsing a URL", beginning on page 499 of their book, describe and illustrate several methods that are part of the URL class that help one parse (or split) a URL to its various components. They include sample code that shows what is returned by various methods on the URL 'http://java.sun.com:80/Series/Tutorial/intro.html#DOWNLOADING'. The 'URL.getHost' returns the host name field of a supplied URL ("java.sun.com"). The URL.getProtocol returns the protocol of a supplied URL ("http"). The URL.getPort return the port number of a supplied URL ("80"). The URL.getFile returns the file path of a supplied URL ("/Series/Tutorial/intro.html"). The URL. getRef returns the anchor of a supplied URL ("DOWNLOADING") [Campione-1, page 499-500]. Their example program on page 499 also show how URLs are constructed from its parts through the use of the '+' operator that concatenates two strings together. For example, they show that they constructed the above URL by concatenating http://java.sun.com:80/Series/ + "Tutorial/intro.html#DOWNLOADING". The exact same operators would work on variables containing URL parts as well as the literal strings shown in this example.

c. US Patent 6,594,687 by Praitis

122. In 1999, Praitis et. al. from Microsoft Inc. filed US Patent 6,594,687. This patent describes a method that creates 'friendly error pages' for end users, where the method substitutes a more informative 'friendly' page for the standard error page [US Patent 6,594,687]. One of their techniques generates a substitution page that includes a link to the "home page related to the web server that was initially accessed", i.e., the domain name. As shown in [US

Patent 6,594,687, Figure 9] the new page would then include a message telling the end user that they could open that 'home page' as specified by the domain name, and then look for links to the information they wanted. Finding this 'home page' involves parsing the URL, which one skilled in the art could do by trivially applying the standard string manipulation techniques similar to those mentioned in the PHP section above. As Praitis et. al. writes:

> In order to create the hyperlink to the related home page, the HTML document comprises scripted code that parses the requested URL. In parsing the URL, the code searches for a first predetermined delimiter and disregards all information before the delimiter. Next, the code stores each element of the URL up to a second delimiter and disregards all information following the second delimiter. The information between the two delimiters is used to construct a URL which becomes the requested page should the hyperlink become activated. The constructed URL comprises all necessary information such as the proper protocol, delimiters, etc. [US Patent 6,594,687, 13:32]

d. US Patent 5,761,436 by Nielsen

123. Jacob Nielsen from Sun Microsystems filed US Patent 5,761,436 in 1996. The patent concerned itself with a history system for hypertext users. It included a method for truncating hyperlinks as part of the way it created a history list, that is, "The method first creates a plurality of truncated hyperlink references by determining a truncated hyperlink for each of the plurality of hyperlinks." [US Patent 5,761,436, col. 2:37]. The process is described in detail in the patent, and for brevity is not reproduced here. Amongst other things, his method can discover and parse out the parent URL (if one exists) shared by several URLs, and the parent of a particular page. Again, one skilled in the art could do by trivially applying the standard string manipulation techniques similar to those mentioned in the PHP section above. To illustrate, we turn to an example provided by Nielsen:

> "This process is described in FIG. 9 and results in a truncated hyperlink specifying a hierarchical reference that is one step removed from a specific hypernode. (That is, in the WWW instance, "http://www.sun.com/netra-nfs/features.html" is truncated to "http://www.sun.com/netra-nfs/")". [US Patent 5,761,436, col. 2:37-39]

e. Chen et. al.'s Cha-Cha

124. On October 11-14, 1999, Chen et. al. from the University of California, Berkeley, published and presented a paper describing the Cha-Cha system for organizing Intranet search results [Chen-1]. They describe its basic functionality as follows.

> Cha-Cha imposes an organization on web site search results by recording the shortest paths, in terms of hyperlinks, from server root pages to every web page within the intranet. After the user issues a query, these shortest paths are dynamically combined to form a hierarchical outline of the context in which the search results occur. This outline structure shows the home pages of the servers on which the search hits occur, as well as the titles of the hyperlinks between the home pages and the search hit. [Chen-1, page 3]

In effect, Cha-cha discloses four ideas pertinent to the '904 patent. First, they provide a method for determining a particular type of home page. Second, they use this home page as the primary index to other pages under it. Third, their technique assumes that they would split and trim URLs. Finally, they use a web crawler to create a data store; in their case "the web crawler is given a list of URLs from which to start (e.g., the home page at www.berkeley.edu)." [Chen-1, page 7]. Cha-cha displays search results within a collapsible text+icon outline to show pages under the server root pages. However, replacing the icons with web page visual thumbnails was know at this time, and the system could easily have been altered to do that.

f. Constructing collections of topically related Web resources, by Terveen et. al.

125. We have already mentioned how Terveen et. al., in their auditorium seating visualization system, employ a method that constructs collections of topically related Web resources. They describe how they can calculate the site root page by looking for a common root between multiple URLs, and for merging two URLs from distinct sites if there is a link to a more general host [Terveen-1]. They also present various heuristics for determining if internal pages

also represent distinct sites [Terveen-3]. One skilled in the art would know that implementing the above techniques would use splitting and trimming of the URL in order to analyze various path components, and to construct URLs to create a root page URL. One skilled in the art would likely use techniques similar to those mentioned above in the PHP section.

g. Ahoy! The Homepage Finder, by Shakes, et. al.

126. In 1997, Shakes et. al. describe a technique called dynamic reference sifting, where they illustrated its use in a system called Ahoy! that lets an end-user search for the home page of a person [Shakes-1]. “Given a person’s name and institution, Ahoy! filters the output of multiple web indices to extract one or two references that are most likely to point to the person’s homepage” [Shakes-1, Abstract]. A vital part of Ahoy! is its URL Pattern Extractor, which “extracts general patterns from URLs found in successful searches, and records them for future references”, and its URL Generator that “relies on extracted patterns to automatically synthesize URLs when a search of its references source fails” [Shakes-1, Section 3]. It then provides examples of how domain names were extracted (or trimmed) from a URL (e.g., emarld.tufts.edu), how parts of the path were looked for (e.g., ~[username] and people/[username]. It then detailed how it would generate possible home pages by joining these path parts to possible user names of a person (e.g., www.cs. tufts.edu/~rjacob, and www.cs.tufts.edu/robjacob/). Thus Shakes et. al. anticipates how trimming and joining can be used to produce possible home page URLs.

h. Summary

127. The above background is an illustrative rather than exhaustive overview of how splitting, trimming, and constructing URLs has been known and used by many others in ways that anticipate various inventive elements of the '904 patent. We have seen that:

- Splitting, trimming, and constructing URLs were well within the expected skills of undergraduate computer scientists (e.g., as shown in my course assignment in 1996);
- Programming languages (such as PHP) almost always include string manipulation functions that are specifically designed to split, trim, and construct strings. URLs are, of course, a subset of strings;
- The Java language also includes functions specifically designed to retrieve different parts of a URL;
- Various patents, academic articles and developed systems describe methods that require one to split, trim, and construct URLs;
- Various patents, academic articles, and developed systems describe methods specifically designed to find and/or construct pages different from a given URL, (e.g., US Patent 6,594,687 that finds and constructs a path specifying a home page of a given URL; US Patent 5,761,436 that creates a parent URL shared by several other URLs; Chen et. al.'s Cha-cha [Chen-1] that records the shortest paths from server root pages to every web page; and Shakes et. al.'s Ahoy! [Shakes-1] that generates URLs based on extracted patterns it has seen in other URLs).

D. MULTIPLICITY OF DOWNLOADERS

128. A claimed inventive element in the ‘904 patent appears to be the use of a multiplicity of downloaders to retrieve web pages and embedded objects corresponding to a list of URLs. This is a basic concept known to those skilled in the art as ‘web crawling’ at the time of the ‘904 patent. I note that the provisional application which matured into the ‘904 application disclosed “a crawler” without defining the term, suggesting the terminology was well known in the art at the time.

a. Pinkerton’s WebCrawler

129. Web crawlers were known and built in the early days of the world wide web usually for the purpose of collecting and indexing its contents. An example web crawler was presented by Brian Pinkerton in his 1994 paper: “Finding What People Want: Experiences with the WebCrawler” [Pinkerton-1]. We use this example to illustrate basic concepts of web crawlers that would be known to those skilled in the art at the time of the ‘904 patent.

130. Pinkerton explains that his web crawler uses a multiplicity of downloaders, as well as why this is a good way to do this. He calls his downloaders ‘agents,’ which are invoked by his search engine component. Figure VI-29 below reproduces Figure 1 in his paper, where we see a multiplicity of agents. Pinkerton explains:

> “To actually retrieve documents from the Web, the search engine invokes "agents." The interface to an agent is simple: "retrieve this URL." The response from the agent to the search engine is either an object containing the document content or an explanation of why the document could not be retrieved. The agent uses the CERN WWW library (libWWW), which gives it the ability to access several types of content with several different protocols, including HTTP, FTP and Gopher [Berners-Lee].

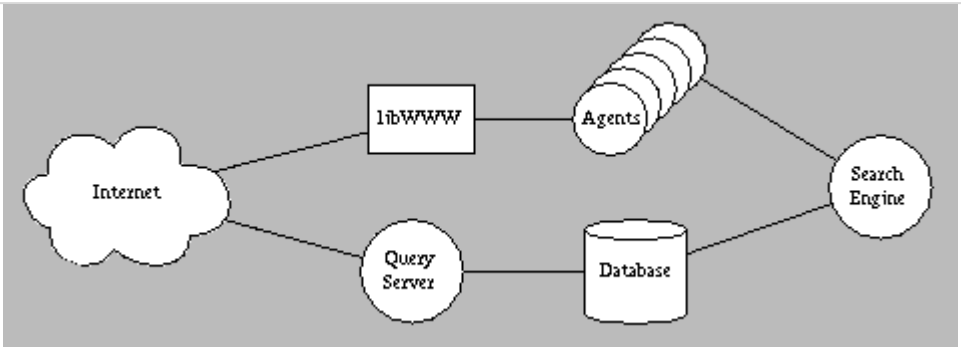


Figure VI-29: Software components of the WebCrawler, Figure 1 from [Pinkerton-1]

> Since waiting for servers and the network is the bottleneck in searching, agents run in separate processes and the WebCrawler employs up to 15 in parallel. The search engine decides on a new URL, finds a free agent, and asks the agent to retrieve that URL. When the agent responds, it is given new work to do. As a practical matter, running agents in separate processes helps isolate the main WebCrawler process from memory leaks and errors in the agent and in libWWW." [Pinkerton-1]

The above quote also reveals that the search engine is the one that maintains a list of URLs, and after deciding whether it should retrieve the page represented by the URL, it invokes a free agent.

131. Pinkerton also explains that the results of these efforts – the collected documents – are stored in a database. While his work is not directed to images, one skilled in the art would know that web crawlers that index images or web page images would also store these images (or rendered thumbnails of these images) in a database. He states:

> The WebCrawler's database is comprised of two separate pieces: a full-text index and a representation of the Web as a graph. The database is stored on disk, and is updated as documents are added. To protect the database from system crashes, updates are made under the scope of transactions that are committed every few hundred documents. [Pinkerton-1]

b. Greenberg's Assignment 3

132. I previously mentioned how, in 1996, I gave my students in my CPSC 441 class an assignment titled 'Building An Image Web Crawler and a Web Server.' The course

assignment as given out is in [Greenberg-3]. The assignment contained 3 parts, where the assignment specifications says: “The more parts completed, the more marks you get”. Part 1a of this assignment directed students to build ‘a muli-threaded image grabber client for HTML pages. Specifically, it asks them to “Write a program that connects to an URL (a web address) and saves all the images on that page as separate GIF and JPEG files in your directory.” [Greenberg-3, Part 1a]. It directs students to read a web page and construct a list of IMG path URLs, i.e., of images specified on that page that would be retrieved as a separate object. It then tell students that these images should be retrieved in parallel, i.e., through a ‘multiplicity of downloaders’.

> for each figure:
>
> create a new socket connection, retrieve the figure (via repeated reads in the select loop), save each figure in its own file, and (optionally) start xv on it after each file is completely retrieved.
>
> Note: Using the select statement, the program should be reading the html pages and all the figures that it sees in parallel! If you do it as “batch” retrievals, you will be heavily penalized. [Greenberg-3, Part 1a]

Notice that the assignment specifically teaches that downloading of images should be done in parallel, and that it specifically teaches away from ‘batch’ processing of these image URLs sequentially (i.e., where students would be heavily penalized for doing this).

133. In addition, the above quote states that students were asked to store any found images as a file. It would have been well within their knowledge abilities to have stored these images in a database to construct an image database (as database courses were also available to 3rd and 4th year students); that I did not do so was simply a matter of avoiding extra work for students in this assignment. Indeed, in part 2 of the assignment I asked them to construct a type of image server, where the image server would return these stored images to the requesting client, as described previously.

134. Part 1b of the assignment asked students to turn this image grabber into a simple web crawler. The web crawler had to create lists of all URLs found in a page associated with a single URL, as well as all URLs found in its children page up to a maximum value. The assignment specifically directs students to:

> look for HTML directives of the form: <a href="http://www.ucalgary.ca/local/UCServers.html">
> -build a list of these pages
> -after the current page is complete, do the same for the pages added to the list (this can be done by adding an outer loop to your image_grabber program); you can do it in parallel, but its easier sequentially. [Greenberg-3]

In other words, students were asked to build a list of URLs, and then use that list to operate a multiplicity of downloaders to retrieve images from those URLs.

135. Thus in 1996, well before the filing date of the '904 patent, I expected students to be able construct a list of URLs, and operate a multiplicity of downloaders on those URLs where it specifically retrieved and stored images. I also note that this was only part 1 of 3 parts of this assignment, so it was a basic expectation that all students could do rather than an expectation of only the more advanced or talented students. I do not recollect any student having difficulty doing this.

c. Chen et. al.'s Cha-Cha

136. I previously mentioned how, On October 11-14, 1999 Chen et. al. from the University of California, Berkeley, published and presented a paper describing the Cha-Cha system for organizing Intranet search results [Chen-1], where they analyze links to construct a page akin to a table of contents. Within this context, they created a customized web crawler to crawl the web and create a data store:

> the web crawler is given a list of URLs from which to start (e.g., the home page at www.berkeley.edu). The crawler is restricted to following links that fall only within a set of domains (e.g., all of *.berkeley.edu), while obeying the robots exclusion standard. The crawler mirrors the full text of the web pages onto disk. This is needed to allow for extraction of sentences for page summaries." [Chen-1, page 7]

Chen et. al. clearly describe that their downloader operates on a list of URLs. While Chen et. al. do not explicitly describe that this process is happening in parallel, one skilled in the art at the time would know that this is the most efficient way of constructing a web page. While Chen et. al. do not explicitly describe storing of images, one skilled in the art would know that this is a simple variation of their method.

d. Constructing collections of topically related Web resources, by Terveen et. al.

137. We have already mentioned how Terveen et. al., in their auditorium seating visualization system, employ a method that constructs collections of topically related Web resources [Terveen-1]. Their method includes a crawler that acts on a list of URLs, and that collects, amongst other things, a thumbnail image of a web page and media contents of pages including images.

> The crawler analyzes the content of pages it fetches in order to build profiles of the site content and structure. Profiles include the following data:
> -Title (of the site's root page);
> - A thumbnail image (of the site's root page)'
> - Links to and from other sites'
> - Media contents of pages and sites, including images, audio files, and movie files;
> - Internal pages of the site... we can provide access to these internal pages in the interface, thus offering "shortcuts" to interesting content…"
> [Terveen-3, page 15; also Terveen-1 page 83]

138. Terveen et. al. describe how their crawler acts on a list of URLs:

> *The Algorithm*. We needed a type of Web crawler, which fetches HTML pages, follows (some of the) links found on the pages, and induces sites from pages. Pages

> that are linked to (by already fetched and analyzed pages) are stored on a queue and become candidates for expansion (fetching and analysis). [Terveen-1, page 80]

139. Terveen et al. describes how the harvested data (including thumbnail) of a given web site is stored into a site profile [Terveen-1]. One skilled in the art would know that an efficient way of storing this information would be to use a database.

e. US Patent 6,058,417 (Hess)

140. I previously mentioned how, on November October 13, 1997, Hess et. al. (eBay) filed a patent titled *Information Presentation and Management in an Online Trading Environment* [US Patent 6,058,417]. Hess's patent is directed for online trading environments, where "images are harvested from a plurality of sites" and "thumbnail images are created corresponding to the harvested images" [US Patent 6,058,417, Abstract]. While his patent is not specifically directed at image thumbnails of web pages, he states that his approaches "are equally relevant to other applications in which image data is collected from disparate sources and presented to a user [US Patent 6,058,417, col. 3:65]. Otherwise, his patent contains elements that directly anticipate several of the claimed inventive elements of the '904 patent.

141. Hess describes how the harvester works on a list of URLs:

> "FIG. 7 is a flow diagram illustrating the image harvesting process according to one embodiment of the present invention. At step 710, image location information is retrieved from the listing database 420 for a set of images that will be downloaded concurrently. According to one embodiment, the image location information is a URL." [US Patent 6,058,417, col. 7:42]

142. Hess then describes how this harvesting of these images is done through a multiplicity of downloaders. Step 30 in Fig. 7 of the patent is labeled 'start multiple downloads' and described in the text below:

> At step 730, multiple image downloads are started using a sockets-based interface. Prior to starting the downloads, it may be necessary to attempt a variety of option configurations in order to establish communication with a particular server. [US Patent 6,058,417, col. 7:66]

143. Next, Hess describes how these harvested images are converted into thumbnails.

> "As will described further below, the harvesting process 455 automatically downloads the specified image, converts it to the appropriate format, and scales it to the appropriate size that is appropriate for use with the Gallery presentation mechanism." [US Patent 6,058,417, col. 7:10]

144. Next, Hess describes how these thumbnail images are stored in a database. Figure 4 of the patent shows the databases, generally described in the text below.

> "Importantly, as one feature of the present embodiment, thumbnail images are not stored as individual files; rather, they are stored in an efficient database format that will be described further below." [US Patent 6,058,417, col. 5:42]

f. Kopetzky et. al.'s Visual Link Previews

145. I previously introduced how, in 1999, Kopetzky and Muhlhauser from Austria [Kopetzky-1] described and built a system that created visual previews of links on a web page for view by end users for the purpose of helping them make a "decision about which links to follow and which to ignore" [Kopetzky-1, page 1525]. While they describe various previewing methods, they focus on thumbnail visual images of web pages, which they called thumbnail previews.

146. Kopetzky discloses that he parses a web page to generate a list of URLs, which he will use to retrieve images and generate thumbnails of them.

> The proxy server has the following tasks:
> • analyze the links in a requested HTML document and generate the preview images for all links in the document; [Kopetzky-1, page 1528]

147. Kopetzky then discloses that he uses a multiplicity of downloaders – in this case browsers that retrieve and render the image and thumbnail - to do this image retrieval. He says that "Part of these activities happen in parallel to speed up processing." [Kopetzky-1, page 1529], and then details:

> (4) The parser analyses the document regarding its structure and searches for link information. If link information is found, for each link an internal Web browser will be started. These browsers will be used to generate the preview images. [Kopetzky-1, page 1529]

148. Kopetzky also stresses that these thumbnail images are stored in a cache, but does not provide details on how the cache should be configured. One skilled in the art would know that the storing the thumbnails – the computed link preview images – 'in a cache could be done with a database.

> • cache the requested HTML documents and the computed link preview images for future access;" [Kopetzky-1, page 1528-29]

g. Webseer by Frankel, et. al.

149. In 1996, Frankel et. al., from the University of Chicago described their Webseer system in their paper "Webseer: an Image Search Engine for the World Wide Web". [Frankel-1]. The goal of Webseer was to locate images on the web, and present these images as thumbnails to the end user. Webseer includes a multi-threaded web crawler (a 'multiplicity of downloaders') that analyzes web pages for a list of URLs, crawls the web, downloads images and rendering them as thumbnails, and then storing them in a database. As they describe:

> Techniques for content-based retrieval that work with a hundred, a thousand, or even ten thousand images will not necessarily scale up to the task of indexing all the images on the World Wide Web. There are an estimated 30 million Web pages (HTML documents). Our preliminary experiments indicate there may be about one-third as many images as Web pages, meaning about
> 10 million images to index. Rough calculations suggest that WebSeer's database will be about 1.5 GB, and storing thumbnails will take up about 15 GB of disk space.

> Crawling the Web to index all the images will require downloading them all. Our current multi-threaded Web crawler can download many pages per second, running on a 200 MHz Pentium Pro PC attached to a dual T1 line shared with the rest of the University. …. [Frankel-1, page 18]

They describe these functions in greater detail below:

> With the exception of the URL Server, which is written in Java, all executables are written in C++ and run on a M.S. Windows NT 3.51 platform.
> 1. The WebSeer Crawler crawls the web downloading both HTML pages and images. The crawler is multi-threaded so that the delay downloading pages is spread over multiple threads. Each thread is connected to a database of previously visited (and waiting to be visited) URLs using the ODBC 2.0 database protocol.
> 2. The URL Server is a multi-threaded java application which receives requests to download URLs from the WebSeer Crawler. Separating the URL server from the Crawler application allows us to download pages from multiple machines (with different operating systems) simultaneously. [Frankel-1, page 7]

h. Summary.

150. The above background is an illustrative rather than exhaustive overview of how many others have developed and used a multiplicity of downloaders to retrieve web pages and embedded objects corresponding to a list of URLs in ways that anticipate various purportedly inventive elements of the ‘904 patent. We have seen that:

- The main function of web crawlers, a well known concept to those skilled in the art of the Internet, is to download and store objects from the web;
- Implementing a web crawler by using a multiplicity of downloaders is a well known standard technique motivated by efficiency;
- Various web crawlers act on images and image thumbnails;
- Various web crawlers operate on a list of URLs;
- Web crawlers almost always store their results, often in databases;

- Building such web crawlers were well within the expected skills of junior undergraduate computer scientists (e.g., as shown in my course assignment in 1996).

E. **SUMMARY.**

151. This background just touches upon the rich literature and systems leading up to time period of the '904 patent. Suffice it to say that there is a wealth of research, development, products, and systems that preceded and covered the purportedly inventive elements found in the claims of the '904 patent. Specific claim by claim comparisons and analyses are covered in greater detail in the following section both in text, and in the accompanying claim charts.

VII. NOTES REGARDING ANALYSIS

a. Claim Construction

152. I have reviewed the proposed claim constructions from all of the parties in this case. For the purpose of this report, I take no position on claim construction other than, in my opinion, the prior art is anticipatory and/or obvious under any party's construction. I reserve the right to revisit my opinion should a particular construction be adopted by the Court.

b. Indefiniteness of Claims 44 and 55

153. Claims 44 and 55 require (amongst other things) "trimming a path component based on the consideration of finding the most representative image of a given web page." I am informed that if a claim limitation is purely subjective then it is indefinite. In my opinion, the requirement of "based on the consideration of finding the most representative image of a given web page" is something that would be subjective because "the most representative

image of a given web page" is a matter of personal preference. Even Yuval Yarom, one of the named inventors on the '904 patent, testified to this effect in his deposition (Yarom TR. 433:4-8; 434:19-435:2). Accordingly, because one of ordinary skill in the art would not know how to practice Claims 44 and 55 without understanding the limitations of "most representative image of a given web page," it would be impossible to either practice or design around such claims.

154. Accordingly, while it is my opinion that Claims 44 and 55 are indefinite (and I have been informed that indefinite terms render a claim invalid), I also am of the opinion that Claims 44 and 55 are anticipated and/or rendered obvious by the prior art as discussed below.

c. Priority Date of the '904 Patent

155. I have been informed that without adequate written description support, a claim is not entitled to the benefit of an early priority date. *New Railhead Mfg., LLC v. Vermeer Mfg. Co.*, 298 F.3d 1290, 1294 (Fed. Cir. 2002).

156. It is my opinion that none of the asserted claims of the '904 patent are entitled to the priority date of the provisional application, but instead should be limited to the benefit of the U.S. utility application filing date – November 8, 2000. Each of the independent claims asserted in this case (and therefore also the claims which depend therefrom), recite express requirements that are not disclosed in the provisional application, explicitly or inherently.

157. For example, asserted independent claims 1, 18, 35, and 46 require (among other things) using an "image server" that stores and provides thumbnail visual images of web

pages. The provisional application does not disclose employing an image server, much less a "separate" image server, as required by independent claims 35 and 46. Instead, the provisional simply describes storing images in a database, without any additional detail. I see nothing in the provisional application that would suggest to one skilled in the art that the database must reside on an image server. On the contrary, the provisional specifically mentions that "[a] client program on the end user's host can analyze the page the user views, and display the images in another window," suggesting that the images are handled on the client side, not the server side.

158. I also note that the corresponding independent claims as originally filed did not disclose an image server. As filed, independent claim 1 read: "A method for presenting Internet information to a user comprising: providing to a user a visual image of a web page containing at least one hyperlink; and at least partially concurrently providing a visual image of another web page of at least one web site which is represented by said at least one hyperlink." [SG0000924] and independent claim 21 read: "A system for presenting Internet information to a user comprising: first functionality providing to a user a visual image of a web page containing at least one hyperlink; and second functionality operative at least partially concurrently with said first functionality for providing a visual image of another web page of at least one web site which is represented by said at least one hyperlink." [SG0000927]. Neither of these discloses an image server.

159. Claims 35 and 46 of the '904 patent additionally require "a web server, separated from said image server." This requirement is not disclosed in the provisional application. As discussed above, I do not believe that the provisional application discloses an image server, so these claims are not adequately supported. Even if the provisional could be read to disclose an image server, it certainly does not disclose an image and web server separate from

one another. In fact, at least one of the descriptions in the provisional application of the way the system could be used explicitly discloses no separation: "A search engine or a related link service can add the images to the search results page, or to the related links page, before transmitting the page to the end user." If the images are added by the search engine before the page is sent to the end user, as this passage states, a person of ordinary skill would have understood that server running the search engine would also add the images. Consequently, the web server would not be separated from the server providing the images. None of the other examples given in the provisional application of how the system could be used suggests to me a separate image and web server. I also note that the requirement in claims 35 and 46 of "a web server, separated from said image server" was not added until June of 2004, when claims 41 and 52 were added to avoid prior art purportedly disclosing a proxy server functioning as both a web server and an image server [SG0001051, SG00001054]. Additionally, the provisional application discloses using a proxy server, and does not distinguish between its functionality for serving images and its functionality for serving web pages.

160. I understand that the Court has not yet construed various terms in the '904 patent. From reviewing the materials submitted by Girafa in support of its preliminary injunction motion, it is my understanding that Girafa alleges that independent claims 1 and 18, require a thumbnail image of a "home page" that is different from the linked-to page (so displaying a small image of www.cnn.com when the link provided was to www.cnn.com would not infringe). See, for example, Girafa's Reply Brief in Support of Its Motion for Preliminary Injunction at 12-13. To the extent this interpretation of the claims is adopted, I do not find it is supported by the disclosure in the provisional application, so any claims so interpreted should only receive the November 8, 2000 priority date. In reviewing the first image submitted in the provisional

application, I see that the first matching site is http://www.adrenalin.com and the first image in the left-hand frame appears to be an image of the home page of the www.adrenalin.com website (See Figure VII-1 which reproduces the first image from the provisional patent). Using the Internet Archive, I looked at the www.adrenalin.com website in late 1999 when these drawings were submitted. The link I found at www.adrenalin.com was a larger version of the image displayed in the left-hand column in position one, reproduced here as **Figure VII-2** below. Using the Internet Archive, I navigated to an interior page of this site (reproduced here as **Figure VII-3**), and found that it had a different appearance from the small image in the left-hand column. I moved my mouse over the large "Adrenalin" lettering at the top of the page, and saw that it was a link to the Adrenalin home page, as indicated by the text that appeared near my cursor. When I clicked on this link, I saw a larger version of the image displayed in the left-hand column in position one. This indicates to me that the small image in the left-hand column is simply a scaled down version of http://www.adrenalin.com, the linked-to page, and not to some other web page. Consequently, the drawings submitted with the provisional application do not disclose that the thumbnail image must be of a page other than the linked-to page, as Girafa now appears to be arguing.

Figure VII-1 reproduces the first image from the provisional application.

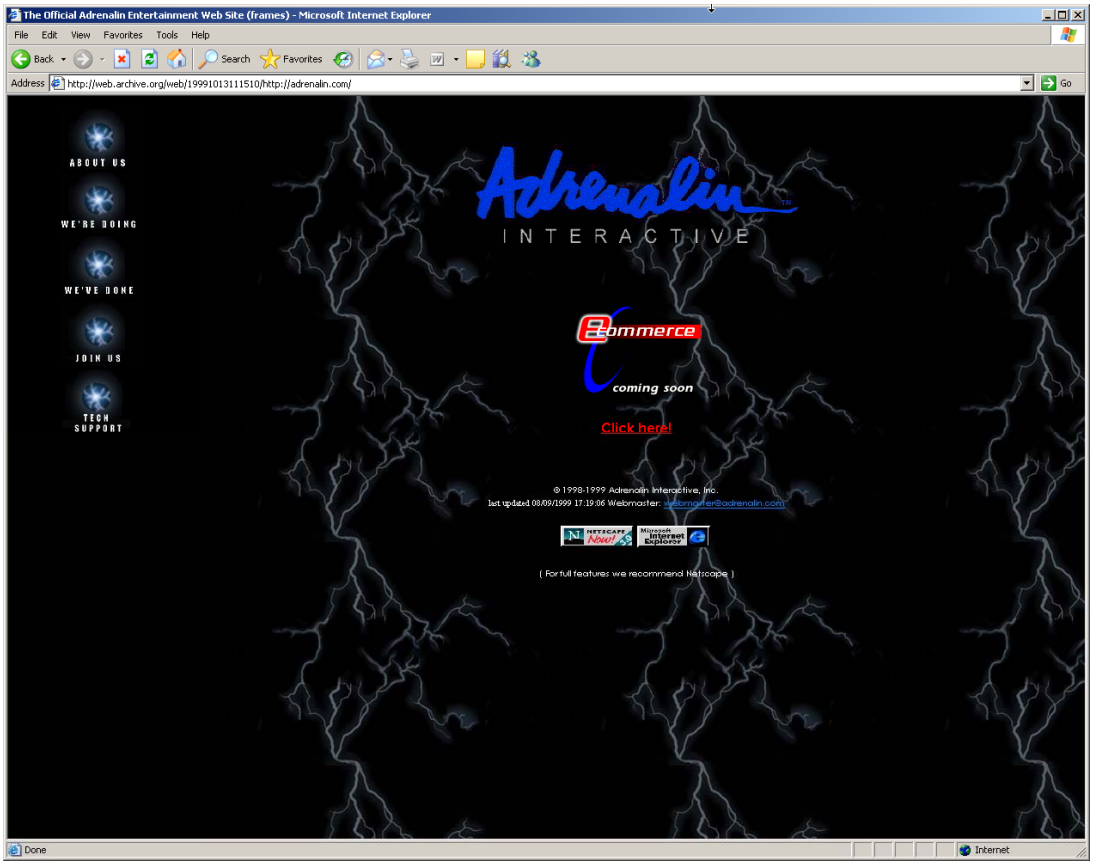


Figure VII-2 is www.adrenaline.com, reproduced from the Internet Archive and dated October 13, 1999.

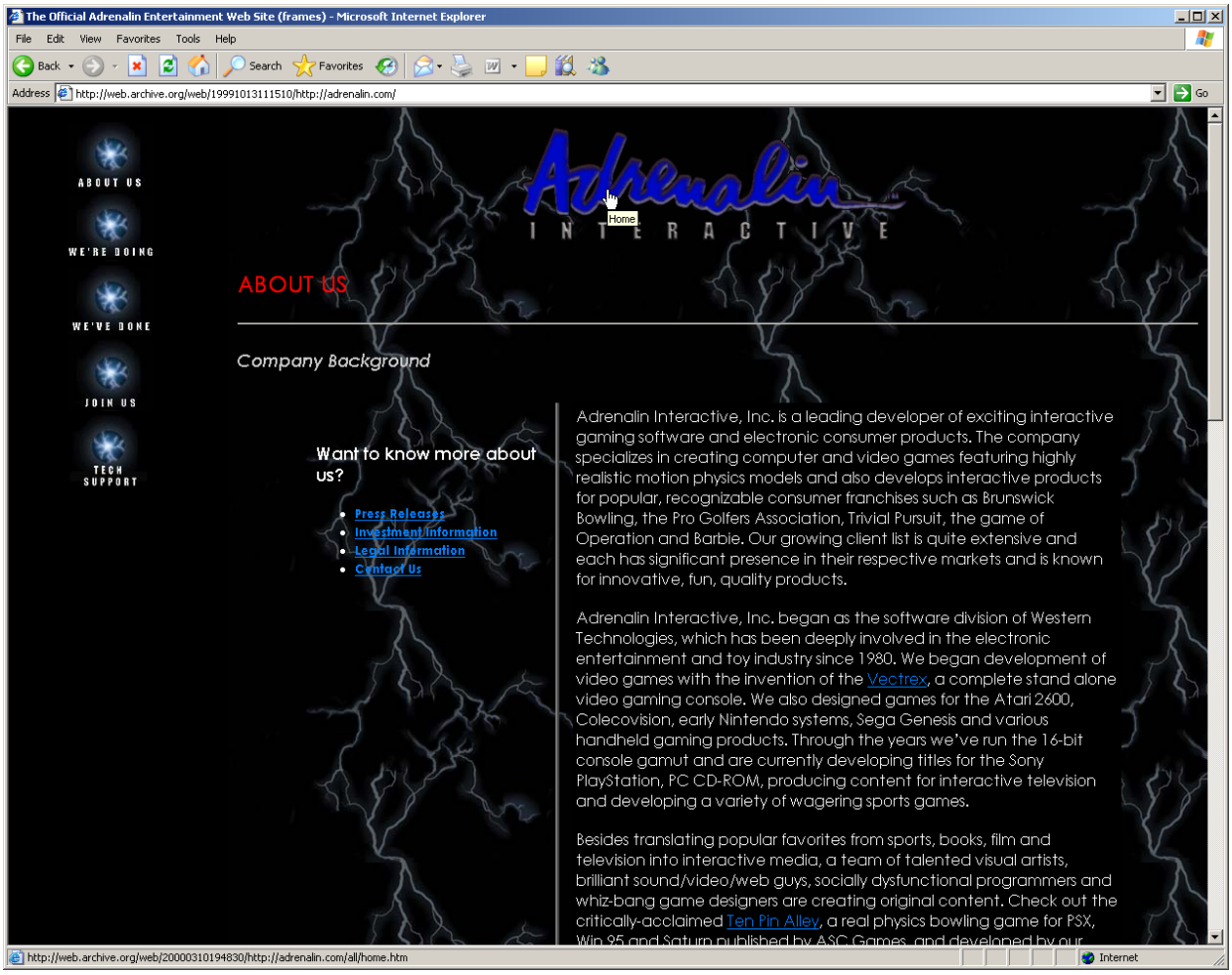


Figure VII-3 is an interior 'About Us' page from www.adrenaline.com, reproduced from the Internet Archive and dated October 13, 1999.

161. Because the claims have not been construed, I do not know if Girafa intends to argue that independent claims 35 and 46 are infringed if something other than the home page of a website is displayed as a thumbnail image. If this argument is made, I note that the provisional only discloses collecting, storing, and providing images of home pages. Collecting, storing, and providing other types of thumbnail images of web sites is not disclosed, so claims 35 and 46 would not be entitled to the provisional priority date, if such a construction is obtained.

162. Certain elements in the dependent claims are not disclosed in the provisional application, above and beyond the non-disclosed elements of the relevant independent claims. Claims 5, 15, 22, 32, 45, 50, and 56 all require that the thumbnail visual image appear hovering over the hyperlink. The provisional application makes no mention of this functionality. Furthermore, claims 12, 13, 18, 29, 30, 35, 44, 46, and 55, all require (in some form) visualization functionality that is either employed in a web server, the browser, or both.[3] With respect to all of these claims, I find no mention of any such visualization functionality in the provisional application and do not believe these claims are entitled to the benefit of the earlier filing date. "Visualization functionality" was not standard terminology at the time the provisional application was filed, and it would not have been apparent to one of skill in the art what was being described, without additional information.

163. Claims 44 and 55 recite a particular algorithm for processing a URL that includes at least splitting the URL into its components, trimming the path component based on finding the most representative image of a given web page, and constructing a new URL. This algorithm is not disclosed in the provisional application. At best, the provisional application simply explains that a URL can be processed by a translation engine to find the key of the image that describes the page. In my opinion, this is insufficient disclosure to support these claims with an earlier priority date.

164. With respect to claims 16 and 33, I believe each of these claims requires operating a multiplicity of downloaders simultaneously to retrieve web pages and embedded content from the Internet. However, the provisional application only discloses using a single

[3] Further, claims 13, 30, 35, and 46 also require embedding commands to the web browser for downloading the thumbnail visual images. I also do not find this concept disclosed in the provisional application.

"crawler" to accomplish this downloading of images, and no reference is made to employing a multiplicity of downloaders. These claims also expressly require the use of a thumbnail generator to create the thumbnail images, yet this disclosure is absent from the provisional application, which does not provide any explanation of what is meant by "thumbnails" as used in the application, much less disclose how to create them. Moreover, as discussed above, the provisional application only discloses a database of images of home pages. To the extent that claims 16 and 33 are read more broadly, to cover images of any web page, this is not disclosed in the provisional application. Finally, with respect to dependent claims 17 and 34, there is no disclosure in the provisional application of deleting executable content from retrieved web pages. Therefore, I also do not believe any of these claims (16, 17, 33, and 34) are entitled to the benefit of the earlier filing date.

165. For the above reasons, in my opinion all of the asserted claims in the '904 patent are only entitled to the November 8, 2000 filing date of the utility application.

VIII. THE '904 PATENT IS ANTICIPATED

166. It is my understanding that a patent claim is anticipated where every element of the claim is found in a single prior art reference before the priority date of the claim. I have been informed by counsel that a patent claim is invalid under 35 U.S.C. §102 if each and every element as set forth in the claim is either expressly or inherently described in a single prior art reference. The prior art reference is said to "anticipate" the claimed invention if it puts the claimed invention in the hands of one skilled in the art. I have been further informed by counsel that although references cannot be combined for anticipation, additional references may be used to interpret the allegedly anticipating reference and shed light on what it would have meant to

those skilled in the art at the time of the invention. These additional references must make clear that the missing descriptive matter in the patent claim is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. I have attached to this expert report element-by-element claim charts of each of the asserted claims in this case with references to the prior art. See Exhibits C-F attached hereto.[4]

A. **PATENT 6,356,908 (BROWN)**

167. US Patent 6,356,908 (Brown) describes an invention and method that presents a page of data contents containing a plurality of links to linked pages by thumbnail visual images, where his patent includes a method for generating and storing these thumbnails. See Historical Background, Sections VII.A.o and VII.B.n.

168. The elements of claims 1, 4-7,12-15, 18, 21-24, 29-32, 35, 38-42, 45, 46, 49- 53, 56 of the '904 patent are present in US Patent 6,356,908. The basis of my opinion is detailed in Exhibit C, an element-by-element claim chart for US Patent 6,356,908. I summarize those opinions below[5].

a. "Thumbnail Visual Image"

169. Brown teaches using thumbnail images to represent a plurality of links to linked pages, where thumbnails are shown within and hovering over a page.

[4] I am also of the opinion that the Miller reference (US Patent 6,613,100) invalidates independent claims 1 and 18, if the Court determines that the thumbnail displayed can be the same as the linked page. This is based on my reading of the file history of the '904 patent.

[5] While I highlight several claimed inventive elements in this and other sections, I am of the opinion that additional elements found in the claims, if any, are also present in the prior art.

170. In addition, Brown teaches thumbnail images of web pages, of which home pages are a natural subset. Brown also teaches that "rather than generating and displaying thumbnails of the web pages associated with links, an icon representing the domain of that link could be generated and displayed next to the text representing the link." In my opinion this icon could, of course, be a thumbnail of the home page, which is consistent with the use of the term in the literature (e.g., Mullet and Sano's book 'Designing Visual Interfaces' [Mullet-1] and US Patent 5,548,692 (Cok), both discussed with regard to Brown in Section VII.A.o.)

b. "Image Server"

171. Brown describes three ways to use an image server to produce visual thumbnails: through a separate protocol/communication with the server, through a proxy server, and via a cache. These variations include a 'separate' image server.

c. Summary

172. In conclusion, US Patent 6,356,908 (Brown) anticipates claims 1, 4-7,12-15, 18, 21-24, 29- 32, 35, 38-42, 45, 46, 49-53, 56 of the '904 patent.

B. **PATENT 7,177,948 (KRAFT)**

173. US Patent 7,177,948 (Kraft) describes an invention and method related to online search technologies that show a short summary description and a visual abstract (i.e., thumbnail image) for each document found in the search. See Historical Background, Sections VII.A.p, and VII.B.o.

174. The elements of claims 1, 4-7, 12-15, 18, 21- 24, 29-32, 35, 38-42, 45, 46, 49-53, 56 of the '904 patent are present in US Patent 7,177,948. The basis of my opinion is

detailed in Exhibit D, an element-by-element claim chart for US Patent 7,177,948. I summarize those opinions below.

a. "Thumbnail Visual Image"

175. Kraft teaches using thumbnail images to represent a plurality of links to linked pages, where thumbnails are shown within and hovering over a page.

176. In addition, Kraft teaches thumbnail images of web pages, of which home pages are a natural subset. Kraft specifically illustrates an example of a thumbnail of a home page representing a web site domain.

b. "Image Server"

177. Kraft teaches using a separate server that renders, stores, and provides thumbnail images from a cache database.

c. Summary

178. In conclusion, US Patent 7,177,948 (Kraft) anticipates claims 1, 4-7, 12-15, 18, 21-24, 29-32, 35, 38-42, 45, 46, 49-53, 56 of the '904 patent.

C. **SCHMID**

179. Stefan Schmid's paper "*Representation with Dynamic Thumbnails*" ('Schmid') [Schmid-1] describes a system and architecture that uses thumbnail visual images to visualize, among other things, the web pages closely linked to the page the user reads. See Historical Background Section VII.A.m. Schmid also describes a client-server thumbnail service

that behaves as an image server that delivers thumbnail visual images to clients. See Historical Background Section VII.B.m.

180. The elements of claims 1, 4, 6, 7, 12-14, 18, 21, 23, 24, 29-31, 35, 38, 40-42, 46, 49, 51-53 of the '904 patent are present in Schmid. The basis of my opinion is detailed in Exhibit E, an element-by-element claim chart for Schmid. I summarize those opinions below:

a. "Thumbnail Visual Image"

181. Schmid teaches thumbnail images of links to represent a plurality of links to linked pages, where thumbnails are shown within a page.

182. In addition, Schmid teaches the use of thumbnails of "any Web pages," of which home pages are a natural subset. In addition, Schmid specifically illustrates an example of a thumbnail of a home page representing a web site domain.

b. "Image Server"

183. Schmid teaches a thumbnail service based on a client-server model, which behaves as a separate image server that renders, stores and provides thumbnail images.

c. Summary

184. In conclusion, Schmid anticipates claims 1, 4, 6, 7, 12-14, 18, 21, 23, 24, 29-31, 35, 38, 40-42, 46, 49, 51-53 of the '904 patent.

D. **KOPETZKY**

185. Kopetzky and Muhlhauser's paper "*Visual preview for link traversal on the World Wide Web*" ('Kopetzky') [Kopetzky-1] discloses a visual link preview – which he also calls thumbnail previews – to present Internet information for view by end users for the purpose of helping them make a decision about which links to follow and which to ignore. Kopetzky also describes an architecture, including a separate thumbnail image server, that collects, renders, and stores thumbnail visual images (via a multiplicity of downloaders) so they can be delivered to the client on demand. See Historical Background Section VII.A.l, VII.B.k, and VII.D.f.

186. The elements of claims 1, 4-7, 12-18, 21-24, 29-35, 38-42, 45, 46, 49-53, 56 of the '904 patent are present in Kopetzky. The basis of my opinion is detailed in Exhibit F, an element-by-element claim chart for Kopetzky. I summarize those opinions below.

a. "Thumbnail Visual Image"

187. Kopetzky teaches thumbnail images of links to represent a plurality of links to linked pages, where thumbnails are shown within and hovering over a page

188. In addition, Kopetzky teaches thumbnail images of web pages, of which home pages are a natural subset. Kopetzky specifically illustrates an example of a thumbnail of a home page representing a web site domain.

189. In addition, Kopetzky teaches that the thumbnail image may represent a location other than the one pointed to in the link, where he illustrates how the URL of the link would retain the #hypertext anchor, yet be represented by a thumbnail image that was generated from the URL without the hypertext anchor.

b. "Image Server"

190. Kopetzky teaches a separate thumbnail image server that collects, renders, and stores thumbnail visual images so they can be delivered to the client on demand.

c. "Multiplicity of downloaders"

191. Kopetzky teaches receiving and analyzing a web page to extract a list of URLs, and generating and storing thumbnail images of the pages for these links.

192. In addition, Kopetzky teaches a multiplicity of downloaders by starting an internal web browser for each link, and using that browser to generate the preview images. He explicitly states that doing these activities in parallel speeds up processing.

d. Summary

193. In conclusion, Kopetzky anticipates claims 1, 4-7, 12-18, 21-24, 29-35, 38-42, 45, 46, 49-53, 56 of the '904 patent.

IX. GIRAFA'S COMBINATION PATENT IS OBVIOUS

194. I am not an attorney. I have, however, read the recent Supreme Court decision in the matter of *KSR Int'l Co. v. Teleflex, Inc.*, 127 S. Ct. 1727 (2007). I understand that in *KSR* the Supreme Court elaborated upon the framework for analyzing obviousness it had set forth in previous cases including *Graham v. John Deere.* It is also my understanding that in *KSR*, the Supreme Court rejected the Federal Circuit's "rigid" application of the "teaching, suggestion,

or motivation" test for obviousness in favor of an "expansive and flexible approach" using "common sense." *KSR,* 127 S. Ct. at 1739.

195. I also understand that in its opinion in *KSR*, the Supreme Court specifically cautioned against granting patents that are nothing more than combinations of known elements driven by non-innovative factors such as market demands. *Id.* at 1732. The Supreme Court in *KSR* also provided guidance on how combination patents like those at issue in this case should be handled. The Court noted that "[g]ranting patent protection to advances that would occur in the ordinary course without real innovation retards progress and may, in the case of patents combining previously known elements, deprive prior inventions of their value or utility." *Id.* I understand that the Supreme Court stressed the need for "caution" before validating patents that are merely combinations of elements found in the prior art. *Id.* at 1741. In view of this caution, the Supreme Court explained that "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *Id.* at 1731.

196. I further understand that the Court pointed to other factors which may show obviousness. For example, the Supreme Court observed, "[w]hen a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill in the art can implement a predictable variation," it is obvious. *Id.* And "[i]f a technique had been used to improve one device, and a person of ordinary skill would recognize that it would improve similar devices in the same way, using the technique is obvious, unless its actual application is beyond his or her skill." *Id.* Further, "[w]hen there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical knowledge." *Id.* at 1732. Also, "[i]f a

person of ordinary skill can implement a predictable variation of the prior art in the manner claimed, §103 likely bars its patentability." *Id.* at 1740.

197. I understand that in *KSR*, the Supreme Court also stated that the factors articulated in *Graham* are to be used in the obviousness analysis. These factors are: (1) the scope and content of the prior art, (2) differences between the prior art and the claims asserted, and (3) the level of ordinary skill in the pertinent art. *Graham v. John Deere Co., of Kansas City*, 383 U.S. 1, 17 (1966). The Supreme Court indicated analyzing "secondary considerations" may also be done, but is not required. *Id.* at 17-18.

A. GIRAFA'S PATENT IS THE TYPE OF "COMBINATION PATENT" ADDRESSED IN KSR

198. The '904 patent is simply a combination of known elements, to the extent any claim is not anticipated by a single prior art reference. Yuval Yarom, a named inventor on the '904 patent, testified that Girafa's claimed invention covered several things:

> One of them is showing the thumbnail images of home pages of links appearing in one Web page when showing that Web page. And another thing that we have invented is a certain way of providing – of giving these images, providing these images, the thumbnail images, that are related to links in a Web page. So just the specific way of doing that. And I believe that we also had another – another thing that we have invented there is how to populate the database of images which relates to – to that part – to that method that we have invented. [Yarom Tr. 46:6-47:4]

Yarom acknowledged that Girafa did not invent the idea of showing a thumbnail image of a home page where the link was to that page, did not invent image servers, web servers, or separate image and web servers, did not invent databases or parallel processing, and did not invent downloading web pages to render thumbnail preview images. [Yarom Tr. 49:5-9; 57:19-61:9; 61:21-62:21; 66:10-67:1] Asked to sum up the inventive aspects of the '904 patent, Yarom responded "It's – it's a combination of components that are working together to get the – to get

the end result." [Yarom Tr. 55:16-18] In my opinion, the '904 patent is simply a combination of known elements, all of which were in the prior art.

B. THE COMBINATIONS IN GIRAFA'S PATENT CLAIMS ARE PREDICTABLE AND DO NOT YIELD ANY UNPREDICTABLE RESULTS

199. The Supreme Court stated in *KSR* that "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *KSR*, at 1731. The Supreme Court also stated "[w]hen a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill can implement a predictable variation, §103 likely bars its patentability." *Id.*

200. Here, it is my opinion the combinations were both predictable and produced no unpredictable results.

201. Combining the existing systems for displaying thumbnail images of web pages with a homepage image was not only predictable; it was done. See, for example, the many references in Section VI Historical Background, such as Microsoft's Internet Explorer thumbnail view [Micorsoft-1], CNN Interactive's Site-Seer: The DOJ v. Microsoft [CNN-1], Ayers et al.'s Mosaic-G [Ayers-1], Kopetzky et al's Visual Link Previews [Kopetzky-1], and so on. Even if the "home page" term in the patent is construed to exclude situations where the home page preview displayed is what would be found at the corresponding URL, the combination of prior art disclosing substituting a preview image for an image of the linked to page, as disclosed in Section VI and discussed in detail below as well as in the claim charts accompanying this report – disclosing thumbnail web page preview systems was predictable. For example, Bederson's et

al.'s Pad++ and PadPrints [Bederson-1 and Bederson-2], Cockburn et al.'s WebView [Cockburn-1] and US Patent 6,356,908 (Brown). In my opinion, there is no "unpredictable" result from the combination of a home page, however the term is construed, with the pre-existing systems for displaying thumbnail visual images of web pages. If one of ordinary skill in the art decided to display a home page preview image in lieu of a preview of the linked to page, the result would be perfectly predictable – the home page preview image would be displayed. Thus, Girafa's patent presents no unpredictable results from this combination. Instead, the result is obvious as a matter of common sense, and further supports my opinion that these claims are obvious.

202. Similarly, combining an image server that stored and provided thumbnail visual images of web page with the prior art systems describing separate image and web servers was predictable, to the extent it was not already done. The concept of a separate image server was well known by the time of the '904 patent, and the pros and cons of such a setup were similarly well understood. See, for example, the many references in the VI.B 'Image Servers' Section, which show that the concept of separate image servers was well-understood. Putting thumbnail visual images of web pages on a separate image server would yield no unpredictable results – the preview images would be stored on the separate image server, and would be available to download from the image server. This result is also obvious as a matter of common sense, and further supports my opinion that the claims combining these elements are obvious.

203. Similarly, combining the disclosed systems for generating collections of thumbnail visual images of web pages with the prior art systems for employing multiple downloaders was predictable, to the extent such combinations are not found in a single prior art reference. Running the sort of crawlers with multiple downloaders, as described and justified in various references in VI.D 'Multiplicity of Downloaders,' would not lead to any unpredictable

results – the system would run more quickly because multiple downloaders were being used. Far from being unpredictable, this result is obvious as a matter of common sense, and would have been unremarkable to the average computer science student, as illustrated by the assignment I gave my undergraduates in 1996. As above, this result is a matter of common sense, and further supports my opinion that these claims are obvious.

C. ONE SKILLED IN THE ART WOULD HAVE BEEN MOTIVATED TO PURSUE THE CLAIMED COMBINATIONS THROUGH MARKET FORCES AND TRENDS

204. In *KSR*, the Supreme Court also observed that "when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense. In that instance the fact that a combination was obvious to try might show that it was obvious under §103." *Id.* at 1732.

205. As set forth above, it is my opinion that most of the asserted claims of Girafa's patent are anticipated by the prior art. Clearly, because the Brown, Kraft, Schmid, and Kopetzky references read upon such claims, the authors of such references were motivated to make the same combinations found in the '904 patent. This far exceeds the threshold for one being "motivated" to combine references--they did. The fact that other similar systems were developed contemporaneously, as discussed in the Background section of this report, further supports my opinion that market forces and trends combined to make the claimed invention obvious.

206. Girafa claims its patent solves the problem of providing a practical system of thumbnail preview images to a user. See, for example, Declaration of Shirli Ran in Support of Girafa's Motion for Preliminary Injunction, ¶¶ 9-13. In the late 1990s, someone who wished to implement such a system would have faced a number of constraints. Chief among these were limited bandwidth and the high cost of electronic storage space. If one wished to implement the system commercially, the question of how to collect, maintain, and serve thumbnail preview images to a variety of client web sites would also have been immediately apparent. The range of solutions available to implement a commercial solution for providing thumbnail preview images to a user was limited. Clearly such a system would require a) a way to collect the images, b) a way to store the images, and c) a way to transmit the images. The obvious solution to the collection problem was a web crawler, and a standard web crawler would employ multiple downloaders for efficiency in light of bandwidth constraints. That this option was in the prior art is clear from the various references in VI.D 'Multiplicity of Downloaders' – for example, as done by Terveen's system [Terveen-1], US Patent 6,058,417 (Hess), and Frankel's Webseer [Frankel-1]. The obvious solution to the storage and transmission problem was to use a server of some sort. No other option would have been feasible for a commercial system, as a matter of common sense. Surveying the landscape, one of ordinary skill in the art would have recognized that a specialized image server that functioned independently would provide maximal flexibility, just as the Akamai and DoubleClick servers could serve images to multiple client websites. To save on storage costs, a person of ordinary skill in the art had a limited number of options – either reduce the size and/or resolution of each image, or reduce the number of images. Reducing the image size can only be done so far: at some point a thumbnail image becomes so degraded that it is no longer useful. Kopetzky et. al., for example, talk about this tradeoff in their choice of

image sizing and resampling method to produce an image that is small enough to save bandwidth and server space, but still of good enough image quality to be recognizable (Kopetzky-1, p. 1527]. Consequently, there would have been a natural lower limit on the size of the thumbnail images, and their quality. One of ordinary skill would have naturally moved on to option number two – reducing the overall number of images. At this point, in light of references such as those including a homepage that does not link to itself (e.g., [Bederson-1, Bederson-2, Cockburn-1, and US Patent 6,356,908 (Brown)], it would have been obvious to substitute a common preview image for an interior page preview image. This solution would have been particularly simple to implement if the common preview image was of the top level domain, as this would have made it readily apparently which preview should be used for any given URL.

207. These are just some examples of how one of ordinary skill in the art would have been motivated to combine the preexisting art as Girafa has done in its claimed invention. All of the solutions discussed in this section of my report, and those which I discuss in my Background section above, were well within the technical grasp of one skilled in the art as of Girafa's claimed priority date. Because such combinations would have been obvious to try, this further supports my opinion that the asserted claims are obvious.

X. THE GRAHAM FACTORS DEMONSTRATE THAT THE '904 PATENT CLAIMS WHICH MERELY COMBINE KNOWN ELEMENTS ARE OBVIOUS

208. As I indicate above, the Supreme Court in *KSR* instructs that the factors in *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1 (1966) for applying the statutory language of 35 U.S.C. §103 are as follows:

> Under §103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness

or nonobviousness of the subject matter is determined. *Id.* at 17.

A. THE SCOPE AND CONTENT OF THE PRIOR ART

209. The first Graham factor, "the scope and content of the prior art" shows the '904 patent to be obvious. As discussed above in Section IX of my report, each element of claims 1, 4-7, 12-18, 21-24, 29-35, 38-42, 45-46, 49-53, 56 existed in the prior art. Moreover, to the extent any individual element may be found lacking, the references discussed above – alone, or in combination with additional references – render the patent obvious.

The Elements Of the '904 Patent Exist In The Prior Art

210. As discussed above, the elements of the '904 patent exist in the prior art. This section summarizes my conclusions and the basis for such conclusions, which are set forth in more detail in the claim charts attached as Exhibits G1 to G16 (referenced throughout as Exhibits 1-16).

211. As discussed above in Section VIII, I believe: Brown anticipates claims 1, 4-7, 12-15, 18, 21- 24, 29- 32, 35, 38- 42, 45, 46, 49- 53, 56; Kraft anticipates claims 1, 4-7, 12-15, 18, 21- 24, 29-32, 35, 38- 42, 45, 46, 49-53, 56; Schmid anticipates claims 1, 4, 6, 7, 12-14, 18, 21, 23, 24, 29-31, 35, 38, 40-42, 46, 49, 51-53; and Kopetzky anticipates claims 1, 4-7, 12-18, 21-24, 29-35, 38-42, 45, 46, 49-53, 56. To the extent these references are found not to be anticipatory, I believe each one renders the identified claims obvious in light of other references existing in the prior art, including the following representative examples:

- Brown in combination with other references. (Exhibit 1)
- Kraft in combination with other references. (Exhibit 2)
- Schmid in combination with other references. (Exhibit 3)

- Kopetzky in combination with other references. (Exhibit 4)
- Cockburn (Exhibit 5)
- CNN Interactive (Exhibit 6)
- Brown specifically in combination with Kraft, Schmid or Kopetzky (Exhibit 7)
- Berners-Lee (Exhibit 8)
- Hess (Exhibit 9)
- Frankel (Exhibit 10)
- Akamai Freeflow (Exhibit 11)
- Sclaroff (Exhibit 12)
- Praitis (Exhibit 13)
- DoubleClick (Exhibit 14)
- Leighton (Exhibit 15)
- Nielsen (Exhibit 16)

Claim 1/18

212. Claims 1 and 18 are similar independent claims, one a system claim and one a method claim.

213. I have carefully reviewed the scope and content of prior art that would have been known to a person of ordinary skill in the art at the time of plaintiff's alleged priority date relevant to these claims.[6] Claims 1 and 18 are obvious in light of the prior art. Each individual technical element of these claims was known within the prior art and its use in the claims was predictable.

[6] As noted in Section VIII.E.b, I do not believe that the provisional application filed December 6, 1999 discloses elements of claims 1 and 18, and the claims dependent thereon, and, in my opinion, none of these claims are entitled to the priority date of the provisional application.

214. Moreover, each element of the '904 patent was used in the claims according to its known function within the prior art.

215. The claims that depend on claims 1 and 18 are likewise obvious.[7]

216. A chart detailing how Brown (US Patent 6,356,908) as modified by Cockburn, or CNN Interactive renders these claims invalid is attached as Exhibit 1.

217. A chart detailing how Kraft (US Patent. 7,177,948) as modified by Cockburn, or CNN Interactive, or Brown (Exhibit 7) renders these claims invalid is attached as Exhibit 2.

218. A chart detailing how Schmid (*Web Representation)* as modified by Cockburn, or CNN Interactive, or Brown (Exhibit 7) renders these claims invalid is attached as Exhibit 3.

219. A chart detailing how Kopetzky (*Visual Preview*) as modified by Cockburn, or CNN Interactive, or Brown (Exhibit 7) renders these claims invalid is attached as Exhibit 4.

Claims 16/33

220. Claims 16 and 33 are similar independent claims, one a system claim and one a method claim.

221. I have carefully reviewed the scope and content of prior art that would have been known to a person of ordinary skill in the art at the time of plaintiff's alleged priority date

[7] At an attempt at brevity, I have limited the discussion within the text of my report to the asserted independent claims (with the exception of dependent claims 44 and 55). My charts, however, include discussion of dependent claims as relevant.

relevant to these claims.[8] Claims 16 and 33 are obvious in light of the prior art. Each individual technical element of these claims was known within the prior art and its use in the claims was predictable.

222. Moreover, each element of the '904 patent was used in the claims according to its known function within the prior art.

223. The claims that depend on claims 16 and 33 are likewise obvious. Claim 17 depends on claim 16, and claim 34 depends on claim 33.

224. A chart detailing how Kraft (US Patent. 7,177,948) as modified by Hess, or Frankel, or Sclaroff renders these claims invalid is attached as Exhibit 2.

225. A chart detailing how Kopetzky (*Visual Preview*) as modified by Hess, or Frankel, or Sclaroff renders these claims invalid is attached as Exhibit 4.

Claim 35/46

226. Claims 35 and 46 are similar independent claims, one a system claim and one a method claim.

227. I have carefully reviewed the scope and content of prior art that would have been known to a person of ordinary skill in the art at the time of plaintiff's alleged priority date relevant to these claims.[9] Claims 35 and 46 are obvious in light of the prior art. Each individual

[8] As noted in Section VIII.E.b, I do not believe that the provisional application filed December 6, 1999 discloses elements of claims 16 and 33, and the claims dependent thereon, and, in my opinion, none of these claims are entitled to the priority date of the provisional application.

[9] As noted in Section VIII.E.b, I do not believe that the provisional application filed December 6, 1999 discloses elements of claims 35 and 46, and the claims dependent thereon, and, in my opinion, none of these claims are entitled to the priority date of the provisional application.

technical element of these claims was known within the prior art and its use in the claims was predictable.

228. Moreover, each element of the ‘904 patent was used in the claims according to its known function within the prior art.

229. A chart detailing how Brown (US Patent 6,356,908) as modified by Berners-Lee, or Akamai FreeFlow, or DoubleClick, or Leighton renders these claims invalid is attached as Exhibit 1.

230. A chart detailing how Kraft (US Patent 7,177,948) as modified by Berners-Lee, or Akamai FreeFlow, or DoubleClick, or Leighton US Patent 6,108,703 renders these claims invalid is attached as Exhibit 2.

231. A chart detailing how Schmid (*Web Representation*) as modified by Berners-Lee, or Akamai FreeFlow, or DoubleClick, or Leighton renders these claims invalid is attached as Exhibit 3.

232. A chart detailing how Kopetzky (*Visual Preview*) as modified by Berners-Lee, or Akamai FreeFlow, or DoubleClick, or Leighton renders these claims invalid is attached as Exhibit 4.

233. Claims 44 and 55 depend upon independent claims 35 and 46 respectively. A chart detailing how Brown (US Patent 6,356,908) as modified by Praitis or Nielson renders dependent claims 44 and 55 invalid is attached as Exhibit 1. A chart detailing how Kraft (US Patent 7,177,948) as modified by Praitis or Nielson renders dependent claims 44 and 55 invalid is attached as Exhibit 2. A chart detailing how Schmid as modified by Praitis or Nielson renders

dependent claims 44 and 55 invalid is attached as Exhibit 3. A chart detailing how Kopetzky as modified by Praitis or Nielson renders dependent claims 44 and 55 invalid is attached as Exhibit 4.

B. **DIFFERENCES BETWEEN THE PRIOR ART AND THE CLAIMS AT ISSUE**

234. As to the second factor, the "differences between the prior art and the claims asserted," given that each element of the '904 patent exists in the prior art the relevant question is one as to the obviousness of the combination of elements.

235. As discussed above, it is my opinion that the prior art anticipates the claims at issue. Moreover, to the extent differences exist, it would have been obvious to one of ordinary skill in the art to combine the anticipatory references with the references discussed in this section of my report. In sum, any differences between the references and the claims at issue, as discussed in the charts attached hereto, are obvious modifications to the pre-existing art.

C. **LEVEL OF ORDINARY SKILL IN THE PERTINENT ART**

236. The third Graham factor is the level of ordinary skill in the pertinent art. As the Supreme Court recognized in KSR, "[a] person of ordinary skill is also a person of ordinary creativity, not an automaton." *KSR*, at 1742.

237. The skill level of one skilled in the art of Internet methodologies and systems for displaying information received over the Internet circa 1999 varied considerably due to the allure of the 'dot com' boom. However, I generally agree with Dr. Brad Myers, an expert previously relied upon by Girafa, that the level of ordinary skill in the pertinent art would have been (1) a Bachelor's degree in Computer Science or a related degree or equivalent experience

and training, and (2) at least two years of additional experience in Internet technologies or user interface design.

D. THE SECONDARY CONSIDERATIONS SET FORTH IN GRAHAM DO NOT ALTER THE CONCLUSION OF OBVIOUSNESS

238. As I indicate above, I understand that the Supreme Court indicated in *KSR* that secondary considerations may be addressed when relevant. In this case, however, there are no secondary considerations that overcome the obviousness determination, nor is it my understanding that Girafa or Girafa's expert has identified any such factors. I have been informed that Girafa bears the burden of proof on this issue. To the extent Girafa's expert offers an opinion on any secondary considerations of non-obviousness, I reserve the right to supplement this report and/or offer a report in rebuttal of any such opinion.

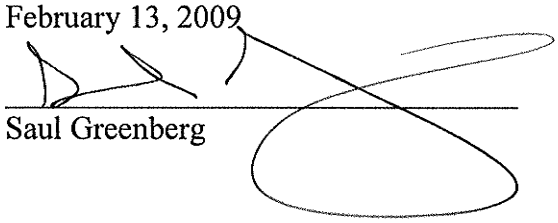
XI. INEQUITABLE CONDUCT

239. I understand that discovery in this case is ongoing, for example, the deposition of Sandy Colb has not occurred. I also understand that defendants continue to seek other discovery from Girafa. I reserve the right to opine on inequitable conduct once discovery is complete and/or should other facts be made available to me.

XII. CONCLUSION

240. It is my understanding that this report is for the purpose of disclosing my opinions and is not meant to be a full recitation of all the details supporting my opinions. If asked, I also intend to testify at deposition and trial. Such testimony would be consistent with the opinions set forth herein.

DATE February 13, 2009

SIGNED 

Saul Greenberg